End User Documentation
Document Version: 1.0 – 2013-01-22

CUSTOMER

SAP InfiniteInsight® 6.5 SP4

Explorer - Text Coding User Guide

Table of Contents

Welcome to this Guide 3

About this Document *3*
- Who Should Read this Document 3
- Prerequisites 3
- What this Document Covers 3
- How to Use this Document 4

Before Beginning *5*
- Files and Documentation Provided with this Guide 5
- Contact Us 6

General Introduction to Scenario 8

Scenario *8*
Introduction to Sample Files *8*
Introduction to InfiniteInsight™ *10*

Extracting Information from Textual Data 11

Simple Method: Using a Classification Model on the Data *11*
- Description 11
- Modeling Process 12
- Results 20

Intermediate Method: Adding Information with the Data Manipulations *23*
- Description 23
- Modeling Process 24
- Results 25

Advanced Method: Using Text Coding to Extract Information from the Textual Variables *28*
- Description 28
- Modeling Process 28
- Results 36

Advanced Method without Stop Words and Stemming Rules *38*
- Description 38
- Modeling Process 39
- Results 39

Adapted Method: Defining a Specific Language for the Domain *41*
- Description 41
- Modeling Process 41
- Results 47

Annex 49

Regular Expression Reminder *49*

Welcome to this Guide

IN THIS CHAPTER

About this Document 3
Before Beginning 5

About this Document

Who Should Read this Document

This document is addressed to the business users who wish to perform tasks using predictive information about their customers or prospects through *InfiniteInsight™* powerful engine. There is no prerequisite for technical data mining knowledge.

Prerequisites

Before reading this guide, you should read chapters 2 and 3 of the *InfiniteInsight™* **- User Guide** that present respectively:

- An introduction to the *InfiniteInsight™*
- The essential concepts related to use of the *InfiniteInsight™* features

When following the scenario described in this user guide, you will have to use KXEN Data Manipulation feature. No prior knowledge of SQL is required to use KXEN Data Manipulation -only knowledge about how to work with tables and columns accessed through ODBC sources. Furthermore, users must have “read” access on these ODBC sources. To use the Java graphical interface, users need write access on the tables `KxAdmin` and `ConnectorsTable`, which are used to store representations of data manipulations.

For more technical details regarding the *InfiniteInsight™*, please **contact us** (on page 6). We will be happy to provide you with more technical information and documentation.

What this Document Covers

This document introduces you to the main functionalities of the *InfiniteInsight™ Explorer - Text Coding* feature. Using the application scenario you can create your first models with confidence.

InfiniteInsight™ Explorer - Text Coding (previously knwon as KTC) lets you build predictive models from data containing textual fields. Thanks to Text Coding models, you can:

- Improve your models with textual processing.
- Handle some text mining problems such as text categorization or mail rerouting.
- Do automatic language recognition.

To know more about the basic concepts underpinning the *InfiniteInsight™*, read the *InfiniteInsight™* **- User Guide**.

How to Use this Document

Organization of this Document

This document is subdivided into three chapters.

This chapter, **Welcome to this Guide**, serves as an introduction to the rest of the guide. This is where you will find information pertaining to the reading of this guide, and information that will allow you to contact us.

The Chapter 2, **General Introduction to Scenario**, provides a summary to the Text Coding application scenario. It also introduces the user interface and the data files used in this scenario.

The Chapter 3, **Standard Modeling with Text Coding**, presents the *InfiniteInsight™ Explorer - Text Coding* feature. It describes how to create **five different predictive models**, by adding data to the original data set, and by using only Classification/Regression for the first two models, and then Text Coding combined with Classification/Regression for the last three models. You will then be able to compare the results obtained with each model.

A summary and detailed table of contents located at the beginning of the guide and cross-references throughout the document allow you to find the information that you need, quickly and easily.

If you want more information on the *InfiniteInsight™* and on the essential concepts of modeling data, read the *InfiniteInsight™* **- User Guide** provided with KXEN software.

Conventions Used in this Document

To facilitate reading, certain publishing conventions are applied throughout this guide. These are presented in the following table.

The following information items...	Are presented using...	For example...
Graphical interface features and file names	**Arial bold**	Click **Next**
The titles of particularly useful sections	Garamond italicized bold	See Operations
The titles of procedures	☑	☑ **To Select the Target Variable**

The titles of sections specific to the scenario presented in this guide		For this Scenario

Before Beginning

Files and Documentation Provided with this Guide

Sample Data Files

Both the evaluation version and the registered version of the *InfiniteInsight™* are supplied with sample data files. These files allow you to take your first steps using various features of the *InfiniteInsight™*, and evaluate them.

During installation of the *InfiniteInsight™*, the following sample files for Text Coding are saved under the folder **Samples\KTC**:

- `dmc2006.txt`
- `desc_dmc2006_without_textual.txt`
- `dmc2006_enriched.txt`
- `desc_dmc2006_enriched_no_textual.txt`
- `desc_dmc2006_enriched_textual.txt`

To obtain **a detailed description of these files** (see "**Introduction to Sample Files**" on page 8).

The folder `Samples\KTC` is located:

- for **Windows**, in the folder `Program Files\KXEN\InfiniteInsight6.1.0\Samples\KTC`
- for **UNIX**, in the folder `Samples\KTC` located in the folder where you have decompressed the **KXENAF** archive file (that is .tar.Z or .tar.gz).

Supported Languages Files

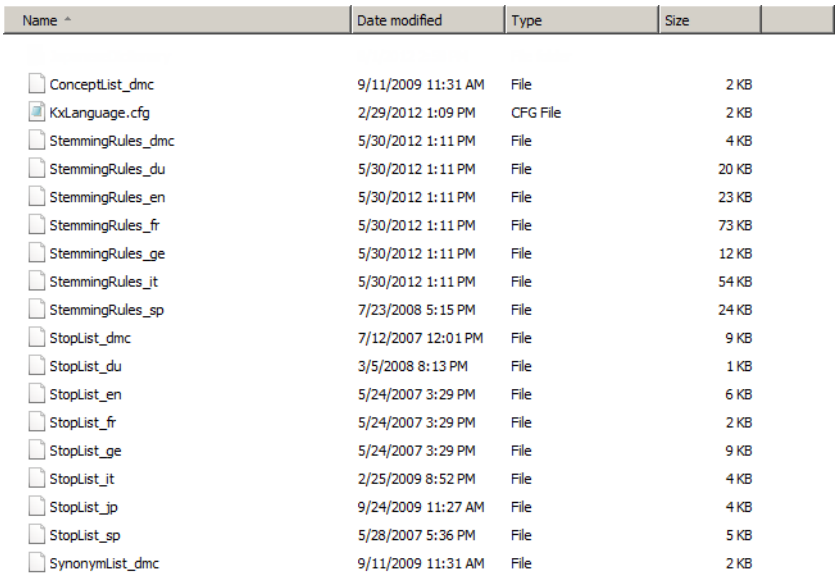
The *InfiniteInsight™ Explorer - Text Coding* feature comes packaged with rules for several languages and can be easily extended to other languages.

The pre-packaged that comes with the installation includes:

- Dutch (Du),
- English (En),
- French (Fr),
- German (Ge),
- Spanish (Sp),
- and Italian (It).

The folder `Resources\KTCData` is located:

- for **Windows**, in the folder `\Program Files\KXEN\InfiniteInsight6.1.0\Resources\KTCData`,
- for UNIX, in the folder `Resources\KTCData` located in the folder where you have decompressed the KXENAF archive file (that is .tar.Z or .tar.gz).



Name	Date modified	Type	Size
ConceptList_dmc	9/11/2009 11:31 AM	File	2 KB
KxLanguage.cfg	2/29/2012 1:09 PM	CFG File	2 KB
StemmingRules_dmc	5/30/2012 1:11 PM	File	4 KB
StemmingRules_du	5/30/2012 1:11 PM	File	20 KB
StemmingRules_en	5/30/2012 1:11 PM	File	23 KB
StemmingRules_fr	5/30/2012 1:11 PM	File	73 KB
StemmingRules_ge	5/30/2012 1:11 PM	File	12 KB
StemmingRules_it	5/30/2012 1:11 PM	File	54 KB
StemmingRules_sp	7/23/2008 5:15 PM	File	24 KB
StopList_dmc	7/12/2007 12:01 PM	File	9 KB
StopList_du	3/5/2008 8:13 PM	File	1 KB
StopList_en	5/24/2007 3:29 PM	File	6 KB
StopList_fr	5/24/2007 3:29 PM	File	2 KB
StopList_ge	5/24/2007 3:29 PM	File	9 KB
StopList_it	2/25/2009 8:52 PM	File	4 KB
StopList_jp	9/24/2009 11:27 AM	File	4 KB
StopList_sp	5/28/2007 5:36 PM	File	5 KB
SynonymList_dmc	9/11/2009 11:31 AM	File	2 KB

Documentation

Full Documentation

Complete documentation is included with the *InfiniteInsight™*. This documentation covers:

- The operational use of *InfiniteInsight™* features,
- The architecture and integration of the *InfiniteInsight™* API,
- The Java graphical user interface: the *InfiniteInsight™* **Modeling Assistant**.

Contextual Help

Each screen in the Modeling Assistant is accompanied by contextual help that describes the options presented to you, and the concepts required for their application.

☑ To Display the Contextual Help

1 Click the **Help** button located on the screen lower left corner.

2 Click the **Previous** button to go back to the original screen.

Contact Us

We are interested in your feedback, and welcome your questions and comments. The following table provides a list of e-mail addresses that you may use to contact us.

If you...	Contact our team in...	Send an email to the following address...
Want more business application information	Marketing	info@kxen.com
Have technical questions related to the integration and use of KXEN products	Support	**United States** support-usa@kxen.com **Canada** **support-ca@kxen.com** **mailto:support-ca@kxen.com** **France** support-fr@kxen.com **United-Kingdom** support-uk@kxen.com **Europe, Middle East and Africa** support-emea@kxen.com
Have comments or questions concerning the KXEN documentation	Documentation	documentation@kxen.com

General Introduction to Scenario

IN THIS CHAPTER

Scenario 8
Introduction to Sample Files 8
Introduction to InfiniteInsight™ 10

Scenario

This scenario demonstrates how to use the *InfiniteInsight™ Explorer - Text Coding* feature for creating a standard model.

The file **dmc2006.txt** is the sample data file that you will use to follow the scenario described in this user guide. It is the contest file from the **Data Mining Cup 2006** (**http://www.data-mining-cup.com/2006/wettbewerb/aufgabe/1165919250/**), which is a German eBay file containing auctions with full conformance with protection of data privacy. The data used in this scenario are online auctions from the category "**Audio&Hi-Fi:MP3-Player:Apple iPod**".

The purpose of this scenario is to **predict for new auctions if the actual sales revenue is higher than the average sales revenue** of the product category.

Introduction to Sample Files

InfiniteInsight™ is provided with sample data files allowing you to evaluate the **Text Coding** feature and take your first steps in using it. The data, or variables, contained in the sample file `dmc2006.txt` are described in the following table.

Variable	Description	Example of Values
auct_id	ID number of auction	An index value
Item_leaf_catedory_name	Product category	A numerical value with two decimals
Listing_title	Title of auction	
Listing_subtitle	Subtitle of auction	
Listing_start_date	Start date of auction	A date in the format such as
Listing_end_date	End date of auction	
Listing_durtn_days	Duration of auction	Specification in days
Listing_type_code	Type of auction (normal auction, multi-auction, ...)	
Feedback_score_at_listing_time	Feedback score by the seller at listing time of auction	An integer value
Start_price	Start price (in EUR)	A numerical value with n decimals

Variable	Description	Example of Values
Buy_it_now_price	Buy-it-now price (In EUR, for buy	A numerical value with n decimals
Buy_it_now_listed_flag	Auction listing with buy-it-now option	1 if the information is true
Bold_fee_flag	Auction listing with boldface	1 if the information is true
Featured_fee_flag	Auction listing as homepage top offer	1 if the information is true
Category_featured_fee_flag	Auction listing as category top offer	1 if the information is true
Gallery_fee_flag	Auction listing with gallery image	1 if the information is true
Gallery_featured_fee_flag	Auction listing with gallery (just in gallery view)	1 if the information is true
Ipix_featured_fee_flag	Auction listing with ipix (Additional, xxl, pic.show, pack)	1 if the information is true
Reserve_fee_flag	Auction listing with reserve price	1 if the information is true
Highlight_fee_flag	Auction listing with background color (in list view)	1 if the information is true
Schedule_fee_flag	Auction listing with determination of start time	1 if the information is true
Border_fee_flag	Auction listing with border	1 if the information is true
Qty_available_per_listing	Quantity of offered articles for multi-auctions	An integer value
Gms	Achieved sales revenue (In EUR)	A numerical value with n decimals (for multi-auctions average price of sold articles)
Category_avg_gms	Average sales revenue (In EUR) of product category (item_leaf_category_name)	A numerical value with n decimals
Gms_greater_avg	0 if gms <= category_avg_gms 1 if gms > category_avg_gms	Target

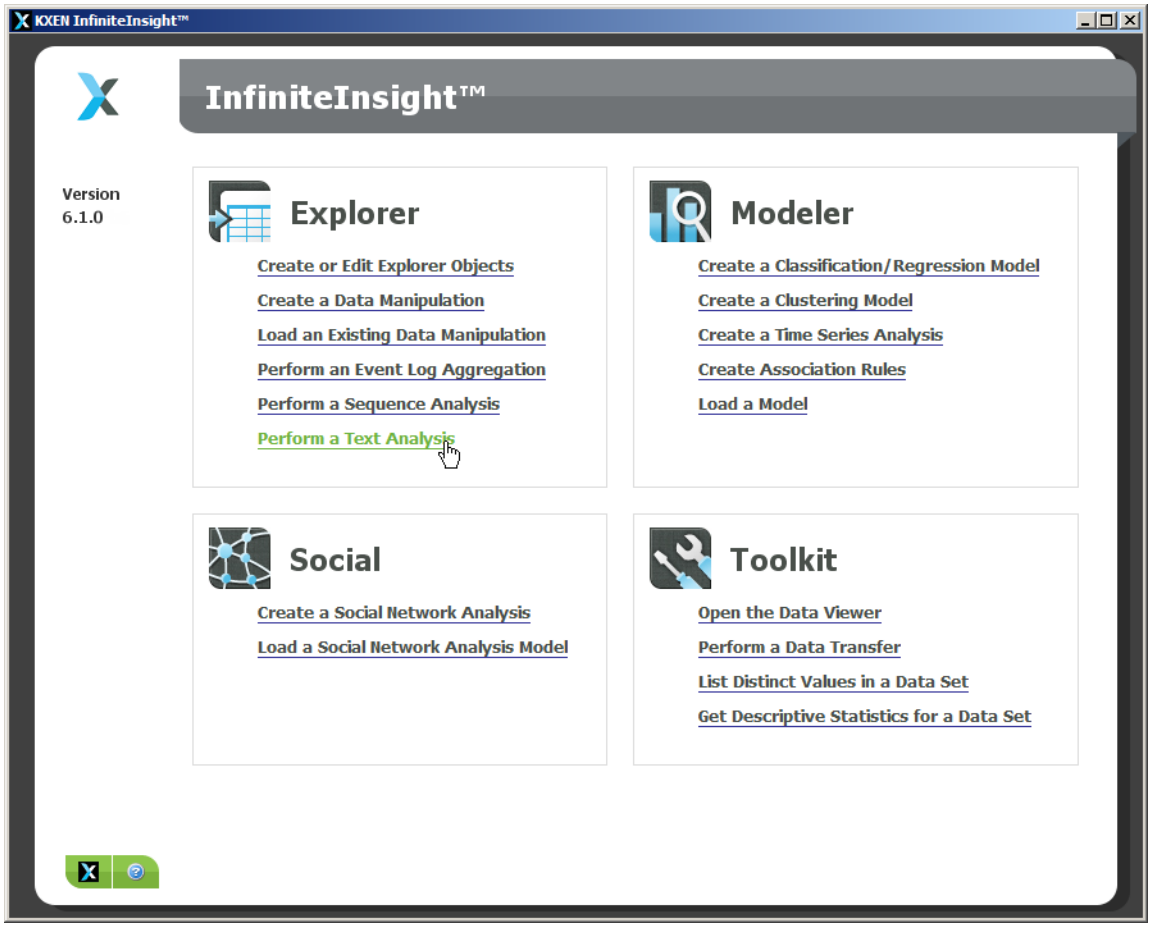
- The file `desc_dmc2006_without_textual.txt` is the description file corresponding to the data file `dmc2006.txt`.
- The file `dmc2006_enriched.txt` is an enriched version of the `dmc2006.txt` data set. The KXEN Data Manipulation feature has been used to create new variables from the ones already existing in the original data set.
- The file `desc_dmc2006_enriched_no_textual` is the description file corresponding to the data file `dmc2006_enriched.txt` with no variable declared as string textual.
- The file `desc_dmc2006_enriched_textual.txt` is the description file corresponding to the data file `dmc2006_enriched.txt` with the **listing_title** variable declared as string textual.

Introduction to InfiniteInsight™

To accomplish the scenario, you will use the Java-based graphical interface of *InfiniteInsight™*. It will allow you to select the feature with which you will work, and lead you through all stages of the modeling process.

☑ To Start the Modeling Assistant

1 Select **Start** > **Programs** > **KXEN** *InfiniteInsight™* > **KXEN** *InfiniteInsight™*.
InfiniteInsight™ welcome page will appear.



2 Click on the feature related to the type of model you want to create in the **Modeler** section.

Extracting Information from Textual Data

IN THIS CHAPTER

Simple Method: Using a Classification Model on the Data 11
Intermediate Method: Adding Information with the Data Manipulations 23
Advanced Method: Using Text Coding to Extract Information from the Textual Variables 28
Advanced Method without Stop Words and Stemming Rules 38
Adapted Method: Defining a Specific Language for the Domain 41

Simple Method: Using a Classification Model on the Data

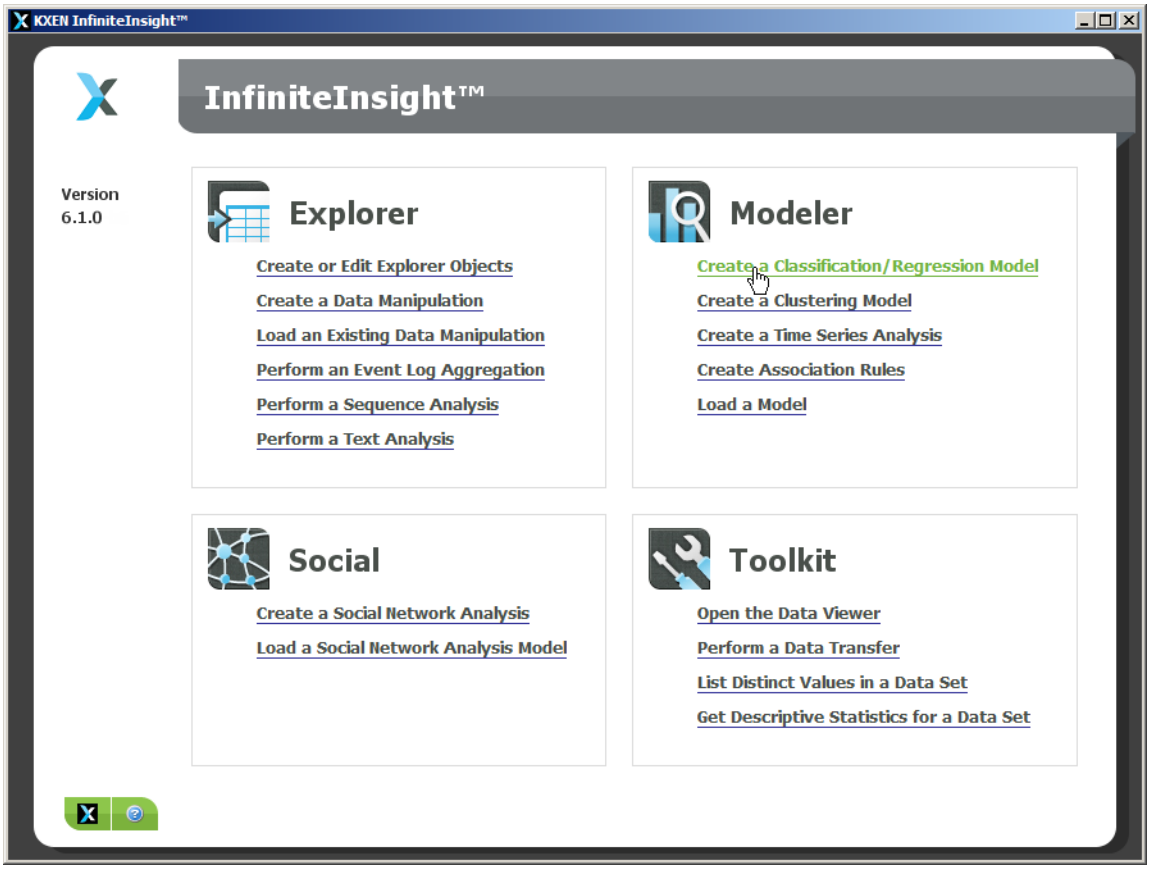
Description

Using *InfiniteInsight™ Modeler - Regression/Classification* feature (previously known as K2R), you will generate a predictive model in order to determine if the auction sales revenue is higher than the sales revenue of its category.

This model will be generated by using as is the data provided in your data base.

☑ To Start a Classification / Regression Model

✓ On the *InfiniteInsight™* start panel, click the option **Classification / Regression** in the **Modeler** Section.



Modeling Process

The *InfiniteInsight™ Modeler - Regression/Classification* feature (*previously known as K2R*) allows you to create explanatory and predictive models.

The first step in the modeling process consists of defining the modeling parameters:

1. Select a **cutting strategy** (see "**Selecting a Cutting Strategy**" on page 15).
2. Select a **data source** (see "**Selecting a Data Source**" on page 13) to be used as training data set.
3. **Describe the data set (see "Describing the Data" on page 16)** selected.
4. Select the **target variable**, and possibly a **weight variable** (see "**Selecting the Target Variable and a Weight Variable**" on page 18).
5. Select the **explanatory variables** (see "**Selecting Explanatory Variables**" on page 19).

Summary of the Modeling Settings to Use

The table below summarizes the modeling settings that you must use for the simple method. It should be sufficient enough for users who are already familiar with the *InfiniteInsight™* Modeling Assistant.

For detailed procedures and more information, see the following sections.

Task(s)	Screen	Settings
▪ Specifying the Data Source ▪ Selecting a Cutting Strategy	**Data to be Modeled**	▪ Select the option **Use a File or a Database**. ▪ In the Folder field, select the folder **Samples/KTC/** ▪ In the Data Set field, select the file **dmc2006.txt**. ▪ Cutting strategy: **Random Without Test**
Describing the Data	**Data Description**	▪ Use the **Analyze** button to obtain the data description.
Selecting the Target Variable and a Weight Variable	**Selecting the Target Variable**	▪ Select **gms_greater_avg** as the target variable ▪ Do not select a weight variable
Selecting Explanatory Variables	**Selecting Variables**	▪ Exclude the variable **gms** from the list of variables to be used for modeling

Selecting a Data Source

After selecting the type of model that you want to generate, you must select:

- The **data source** that you want to use as the training data set.
- A **cutting strategy** to cut your training data set into the three sub-sets: estimation, validation and test.

For more information on Cutting Strategies, see the *InfiniteInsight™* **- User Guide**.

For this Scenario

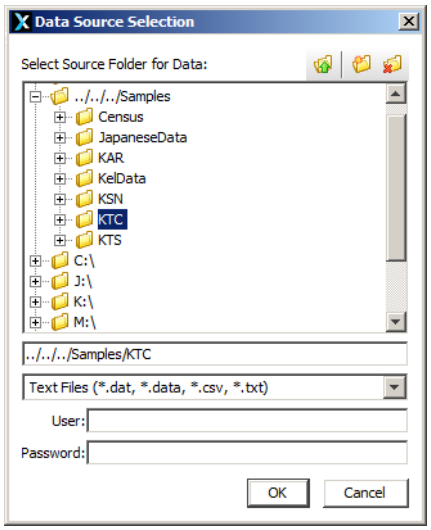
- In the panel **Select a Data Source**, select the options **Use a File or a Database Table** and **Text File** in Data Type.
- In the **Data Set** field, specify the data source to be used by selecting the file **dmc2006.txt**.

☑ To Select a Data Source

1 On the screen **Select a Data Source**, after selecting option **Use a File or a Database Table**, select the option **Text files** in Data Type to select the data source format to be used.

2 Click the **Browse** button.

The following dialog box will appear.

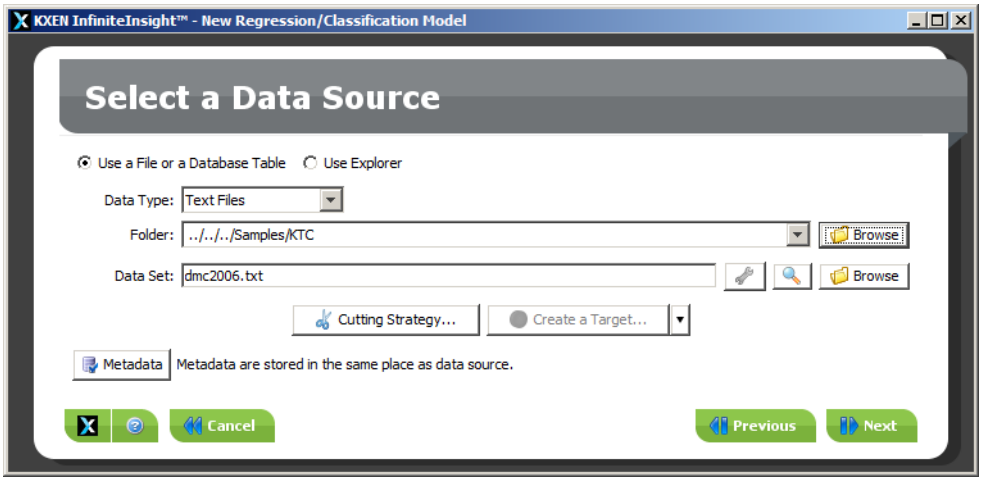


3 Double-click the **Samples** folder, then the **KTC** folder.

Note - Depending on your environment, the **Samples** folder may or may not appear directly at the root of the list of folders. If you selected the default settings during the installation process, you will find the **Samples** folder located in **C:\Program Files\KXEN\KXENCompV3**

4 Select the file **dmc2006.txt**, then click **OK**.

The name of the file will appear in the **Data Set** field.



5 Select a **Cutting Strategy.**

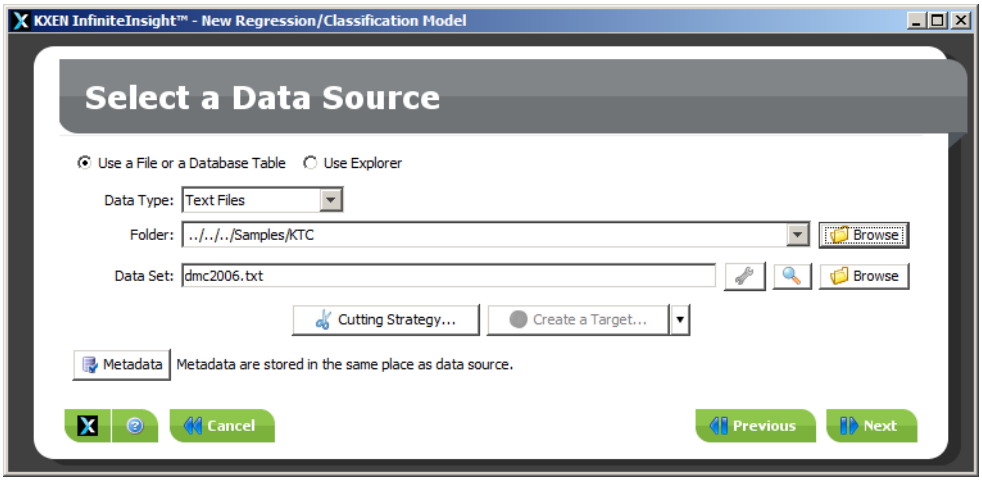
Selecting a Cutting Strategy

For this scenario

Select the **Random without test** cutting strategy.

To Select a Cutting Strategy

1 Once you have selected your Data Source, click on the **Cutting strategy** button.



By default, the **Predifined** mode is set to the **Random without test** cutting strategy.

To get other values, refer to the drop-down list of available cutting strategies.

By selecting the **Custom** mode, you can use the Customized cutting strategy.

2 Click the strategy that you want to use.

Note - In order to use the **Customized** cutting strategy, you must have previously prepared three files corresponding to the three data sub-sets: estimation, validation and test.

3 Click **OK**.

4 Back to the **Select a Data Source** Panel, click the **Next** button.
The screen **Data Description** will appear.

5 Go to the section **Describing the Data Selected**.

Describing the Data

Why Describe the Data Selected?

In order for *InfiniteInsight™* features to interpret and analyze your data, the data must be described. To put it another way, the description file must specify the nature of each variable, determining their:

- **Storage** format: number (**number**), integer (**integer**), character string (**string**), date and time (**datetime**) or date (**date**).
- **Type: continuous**, **nominal**, **ordinal** or **textual**.

> **Warning** - When creating a **text coding model**, you need to define **at least one variable as textual** to be able to go to the next panel.

For more information about data description, see the *InfiniteInsight™* **- User Guide**.

How to Describe Selected Variables

To describe your data, you can:

- Either **use an existing description file**, that is, taken from your information system or saved from a previous use of *InfiniteInsight™* features,
- Or **create a description file using the Analyze option**, available to you in *InfiniteInsight™* Modeling Assistant. In this case, it is important that you validate the description file obtained. You can save this file for later re-use. If you name the description file **KxDesc_<SourceFileName>** , it will be automatically loaded when clicking the **Analyze** button.

> Important - The description file obtained using the Analyze option results from the analysis of the first 100 lines of the initial data file. In order to avoid all bias, we encourage you to mix up your data set before performing this analysis.

Each variable is described by the fields detailed in the following table:

The Field...	Gives information on...
Name	the variable name (which cannot be modified)
Storage	the type of values stored in this variable: ▪ **Number**: the variable contains only "computable" numbers (be careful a telephone number, or an account number should not be considered numbers) ▪ **String**: the variable contains character strings ▪ **Datetime**: the variable contains date and time stamps ▪ **Date**: the variable contains dates
Value	the value type of the variable: ▪ **Continuous**: a numeric variable from which mean, variance, etc. can be computed ▪ **Nominal**: categorical variable which is the only possible value for a string ▪ **Ordinal**: discrete numeric variable where the relative order is important ▪ **Textual**: textual variable containing phrases, sentences or complete texts

The Field...	Gives information on...
Key	whether this variable is the key variable or identifier for the record: ▪ 0 the variable is not an identifier; ▪ 1 primary identifier; ▪ 2 secondary identifier...
Order	whether this variable represents a natural order. There must be at least one variable set as Order in the Event data source. Warning - If the data source is a file and the variable stated as a natural order is not actually ordered, an error message will be displayed before model checking or model generation.
Missing	the string used in the data description file to represent missing values (e.g. "999" or "#Empty" - without the quotes)
Group	the name of the group to which the variable belongs
Description	an additional description label for the variable

For this Scenario

- Create the data description by clicking the **Analyze** button.

☑ To Create a Description File

1 On the screen **Data Description**, click the **Analyze** button.
The data description will appear.

2 Check that the description obtained is correct.

3 Once the data description has been validated, you can:

- Save it by clicking the **Save** button.
- Click the **Next** button to go to the following step.
  The screen **Selecting the Target Variable** will appear.

4 Go to the section Selecting a Target Variable.

A Comment about Database Keys

For data and performance management purposes, the data set to be analyzed must contain a variable that serves as a key variable. Two cases should be considered:

- If the initial data set **does not contain a key variable**, a variable index **KxIndex** is automatically generated by Text Coding. This will correspond to the row number of the processed data.
- If the file **contains one or more key variables**, they are not recognized automatically. You must specify them manually in the data description. See the procedure **To Specify that a Variable is a Key**. On the other hand, if your data is stored in a database, the key will be automatically recognized.

☑ To Specify that a Variable is a Key

1 In the **Key** column, click the box corresponding to the row of the key variable.

2 Type in the value "**1**" to define this as a key variable.

Index	Name	Storage	Value	Key	Order	Missing	Group	Description	Structure
1	auct_id	string	nominal	1	0				
2	item_leaf_category_name	string	nominal	0	0				
3	listing_title	string	textual	0	0				
4	listing_subtitle	string	nominal	0	0				
5	listing_start_date	date	continuous	0	0				

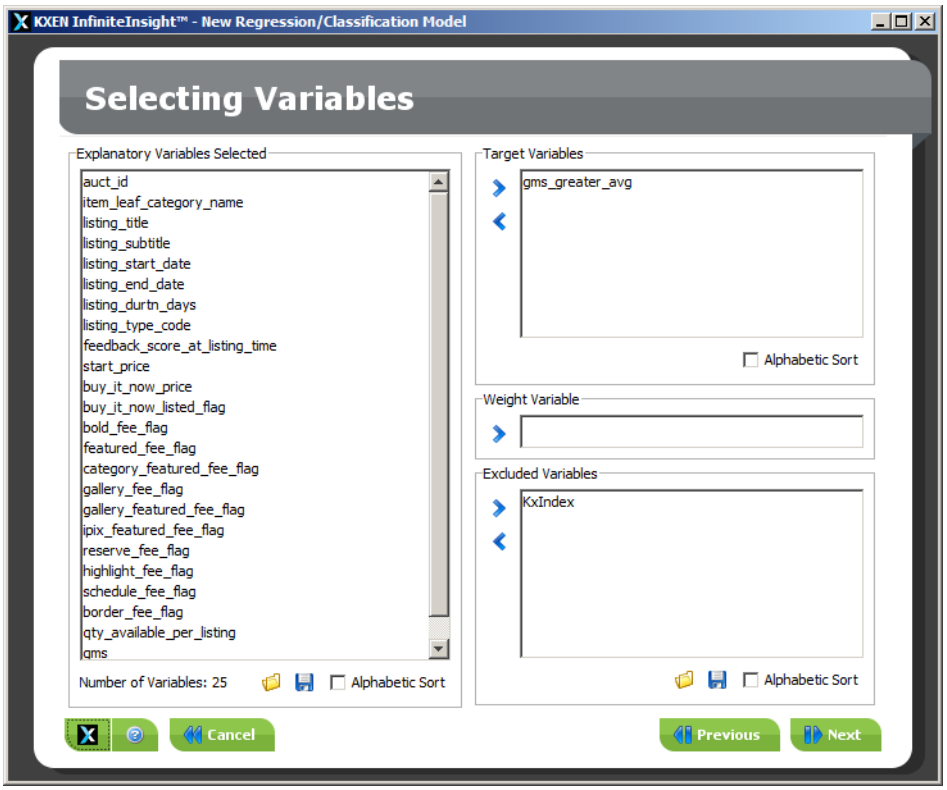
Selecting the Target Variable and a Weight Variable

For this Scenario

- Select the variable **gms_greater_avg** as your target variable.
- Do not select any weight variable.

To Select Target Variable

1 On the screen **Selecting Variables**, in the section **Explanatory variables selected** (left hand side), select the variable you want to use as **Target Variable**.



Note - On the screen **Selecting Variables**, variables are presented in the same order as that in which they appear in the table of data. To sort them alphabetically, select the option **Alphabetic sort**, presented beneath each of the variables list.

2 Click the button **>** located on the left of the screen section **Target(s) Variable(s)** (upper right hand side).

The variable moves to the screen section **Target(s) Variable(s)**.

Also, select a variable in the screen section **Target(s) Variable(s)** and click the button **<** to move the variables back to the screen section **Explanatory variables selected**.

Selecting Explanatory Variables

By default, and with the exception of key variables (such as KxIndex), all variables contained in your data set are taken into consideration for generation of the model. You may exclude some of these variables.

For this Scenario

- Exclude **gms** from the list of variables to be used for modeling the variables since this variable contains the actual amount the auction reached, it answers the question and so would provide a perfect model if used.
- Retain all the other variables.

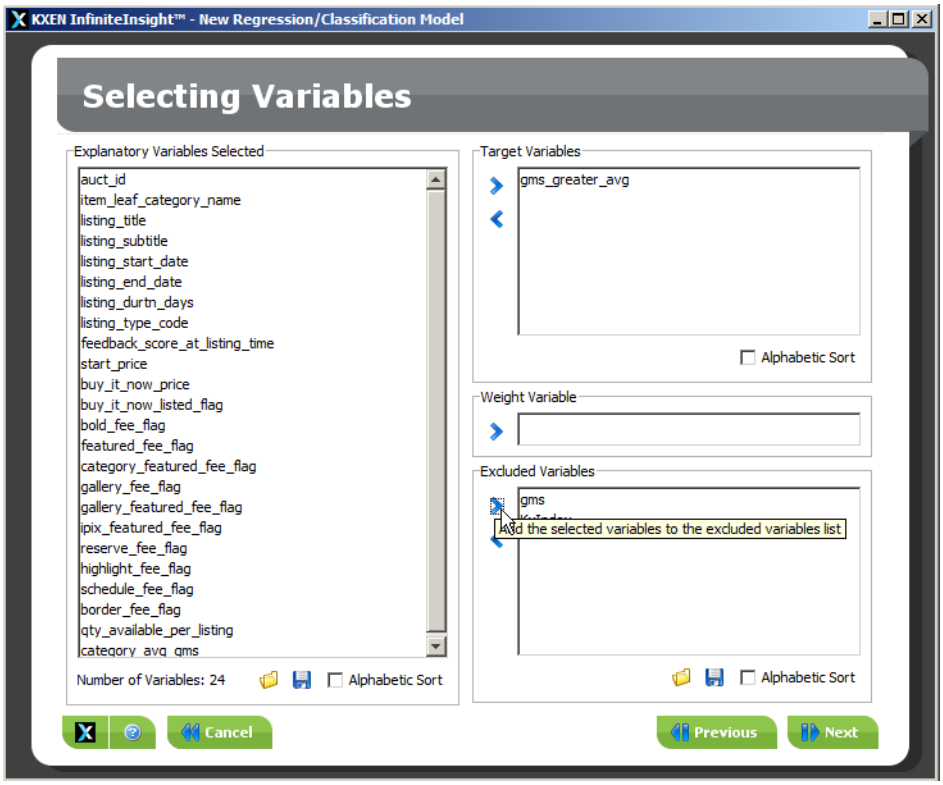
To Select Variables for Data Analysis

1 On the screen **Selecting Variables**, in the section **Explanatory variables selected** (left hand side), select the variable to be excluded.

Note - On the screen **Selecting Variables**, variables are presented in the same order as that in which they appear in the table of data. To sort them alphabetically, select the option **Alphabetic sort**, presented beneath each of the two parts of the screen.

2 Click the button **>** located in the center of the screen. The variable moves to the screen section **Variables excluded**.

Also, click the button **<** to move the variables to the screen section **Explanatory variables selected**.



3 Click the **Next** button. The screen **Summary of the Modeling Parameters** will appear.

Results

Model Performance Indicators

Once the model has been generated, you must verify its validity by examining the performance indicators:

The quality indicator KI allows you to evaluate the explanatory power of the model, that is, its capacity to explain the target variable when applied to the training data set. A perfect model would possess a KI equal to 1 and a completely random model would possess a KI equal to 0.

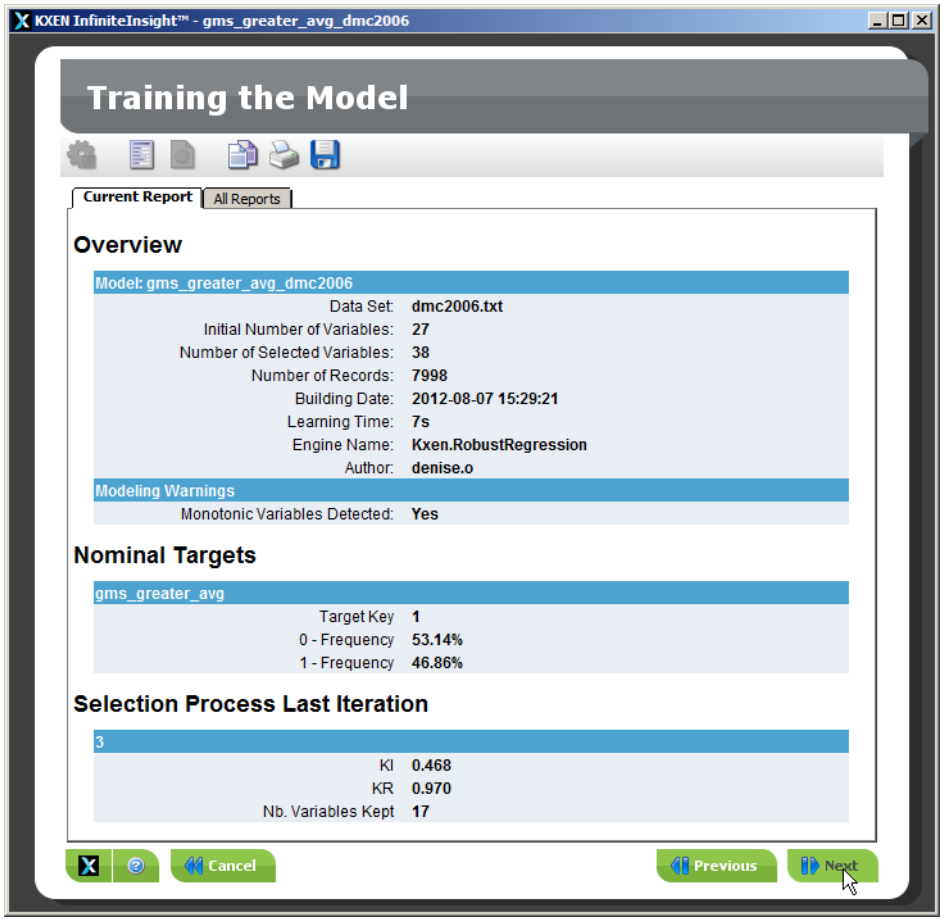
The robustness indicator KR defines the degree of robustness of the model, that is, its capacity to achieve the same explanatory power when applied to a new data set. In other words, the degree of robustness corresponds to the predictive power of the model applied to an application data set.

To see how the KI and KR indicators are calculated, see KI, KR and Profit Curves in the document *InfiniteInsight™* - User Guide.

Note - Validation of the model is a critically important phase in the overall process of Data Mining. Always be sure to assign significant importance to the values obtained for the KI and KR of a model.

The model generated on the provided data gives the following results:

- KI = 0.468
- KR = 0.970



Presentation of the InfiniteInsight™ User Menu

Once the model has been generated, click the **Next** button. The screen **Using the Model** will appear.

The screen **Using the Model** presents the various options for using a model, that allow you to:

- Display the information relating to the model just generated or opened (**Display** section), referring to the model curve plots, contributions by variables, the various variables themselves, HTML statistical reports, table debriefing, as well as the model parameters.
- Apply the model just generated or opened to new data, to run simulations, and to refine the model by performing automatic selection of the explanatory variables to be taken into consideration (**Run** section).
- Save the model, or generate the source code (**Save/Export** section).

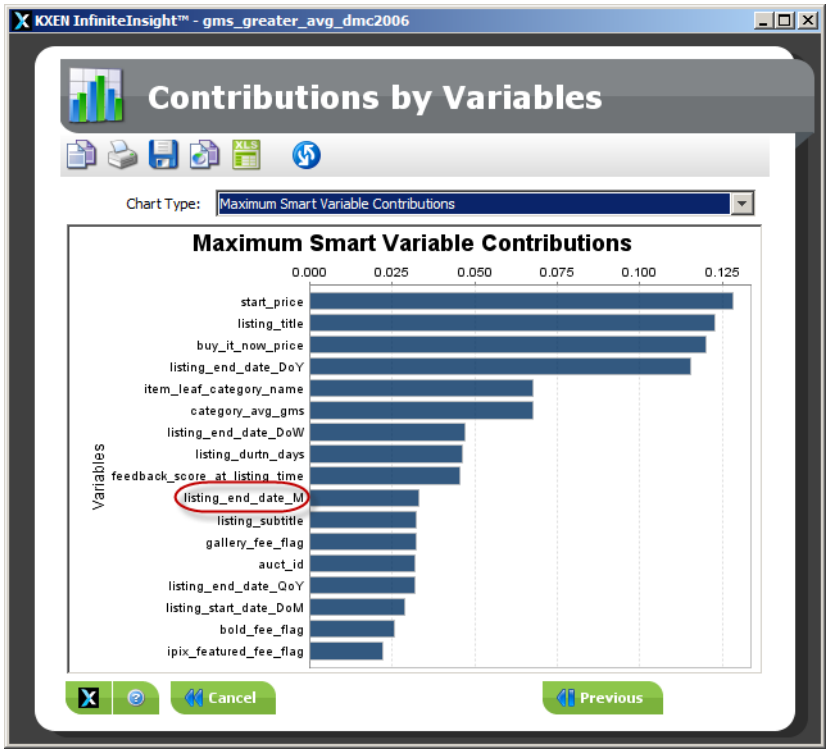
Taking a Closer Look at the Model

From the screen **Using the Model**, you can display a suite of plotting tools that allow you to analyze and understand the model generated in details. The three most useful tools are described in the table below.

On the screen...	You can observe and analyze...
Profit Curves	The performance of the model with respect to a hypothetical perfect model and a random type of model
Contributions by Variables	The contribution of each of the explanatory variables with respect to the target variable
Significance of Categories	The significance of the various categories of each variable with respect to the target variable

On the screen **Contributions by Variables** (see below), you notice that among the variables that contribute the most to the explanation of the target variable is **listing_end_date**. From this result and the knowledge of how auctions work, you can infer that calendar time has an impact on the

auctions results and so you may want to detail this variable content into more informative elements such as the day of the week, the month, and so on. This leads you to the intermediate method.



Intermediate Method: Adding Information with the Data Manipulations

Description

The result of the simple method has highlighted the fact that dates have an important role in the modeling. It seems logical for time information to have an impact on auctions such as the day of the week, the day of the month, the month of the year. You can assume that the results of auctions are better on week-ends or at the beginning of the months or better some months than others, etc…

To make the most of the date variables, you will create new variables, for example by separating the days of the weeks so that they can be used as input in the modeling.

Additionally, to make use of the two other most important variables, you will extract more information from **Start_price** and **Buy_it_now_price** by calculating the ratio between the starting price and the sales mean for the category and the ratio between the By-it-now price and the sales mean for the category.

Modeling Process

The process of building a predictive model on a data set containing added time data is approximately the same as the one you used for building the model on the original data.

The only additional step you have to perform is to create new columns for both variables **listing_start_date** and **listing_end_date**: one for each day of the week, one for the day of the month and one for the month of the year.

The modified data set contains the following added columns:

- extracted from the original variable **listing_start_date**:
- **listing_start_monday**
- **listing_start_tuesday**
- **listing_start_wednesday**
- **listing_start_thursday**
- **listing_start_friday**
- **listing_start_saturday**
- **listing_start_sunday**
- **listing_start_dayofmonth**
- **listing_start_monthofyear**
- extracted from the original variable **listing_end_date**:
- **listing_end_monday**
- **listing_end_tuesday**
- **listing_end_wednesday**
- **listing_end_thursday**
- **listing_end_friday**
- **listing_end_saturday**
- **listing_end_sunday**
- **listing_end_dayofmonth**
- **listing_end_monthofyear**

You will also create two new columns in which the ratios described in the previous section will be stored:

- **Start_price_div_mean_category**, which is the result of the division of **start_price** by **category_avg_gms**.
- **Buy_it_now_price_div_mean_category**, which is the result of the division of **buy_it_now_price** by **category_avg_gms**.

To create these columns, you can use the KXEN Data Manipulation feature. However to speed the process for this demonstration, the modified data set is provided in the folder `Samples/KTC`. The data file that correspond to the original file with the data manipulation creation is `dmc2006_enriched.txt`.

Summary of the Modeling Settings to Use

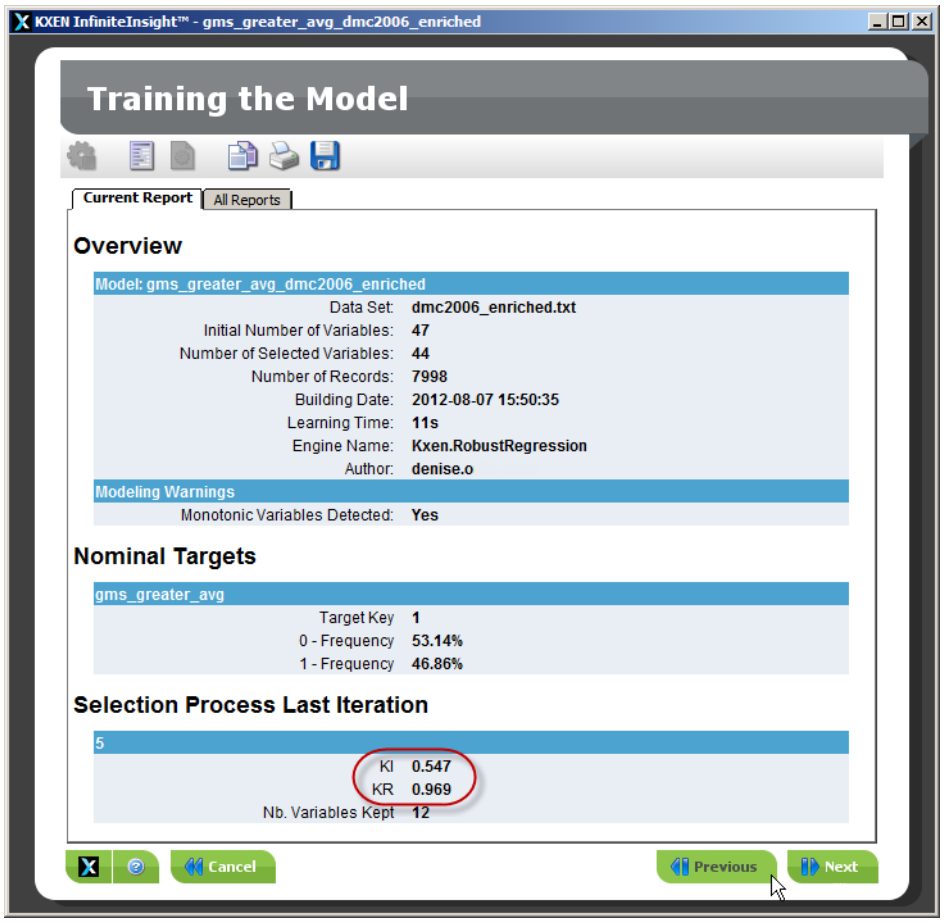
The table below summarizes the modeling settings that you must use for the intermediate method. Except for the additional columns created in the data set, the other settings are similar to the ones used for the simple method.

For detailed procedures and more information, see the Modeling Process section of the Simple Method section.

Task(s)	Screen	Settings
▪ Specifying the Data Source ▪ Selecting a Cutting Strategy	**Data to be Modeled**	▪ Select the option **Text Files** in Datat Type. ▪ In the Folder field, select the folder **Samples/KTC/** ▪ In the Data Set field, select the file **dmc2006_enriched.txt**. ▪ Cutting strategy: Random With No Test
Describing the Data	**Data Description**	▪ Select **desc_dmc2006_enriched_no_textual.txt** as the description file.
Selecting the Target Variable and a Weight Variable	**Selecting the Target Variable**	▪ Select **gms_greater_avg** as the target variable ▪ Do not select a weight variable
Selecting Explanatory Variables	**Selecting Variables**	▪ Exclude the variables **KxIndex** and **gms** from the list of variables to be used for modeling

Results

The screen below shows the quality (KI) and robustness (KR) indicators obtained for the model generated with the additional columns added to the original data set.



The table below compares these results with the ones obtained for the simple method.

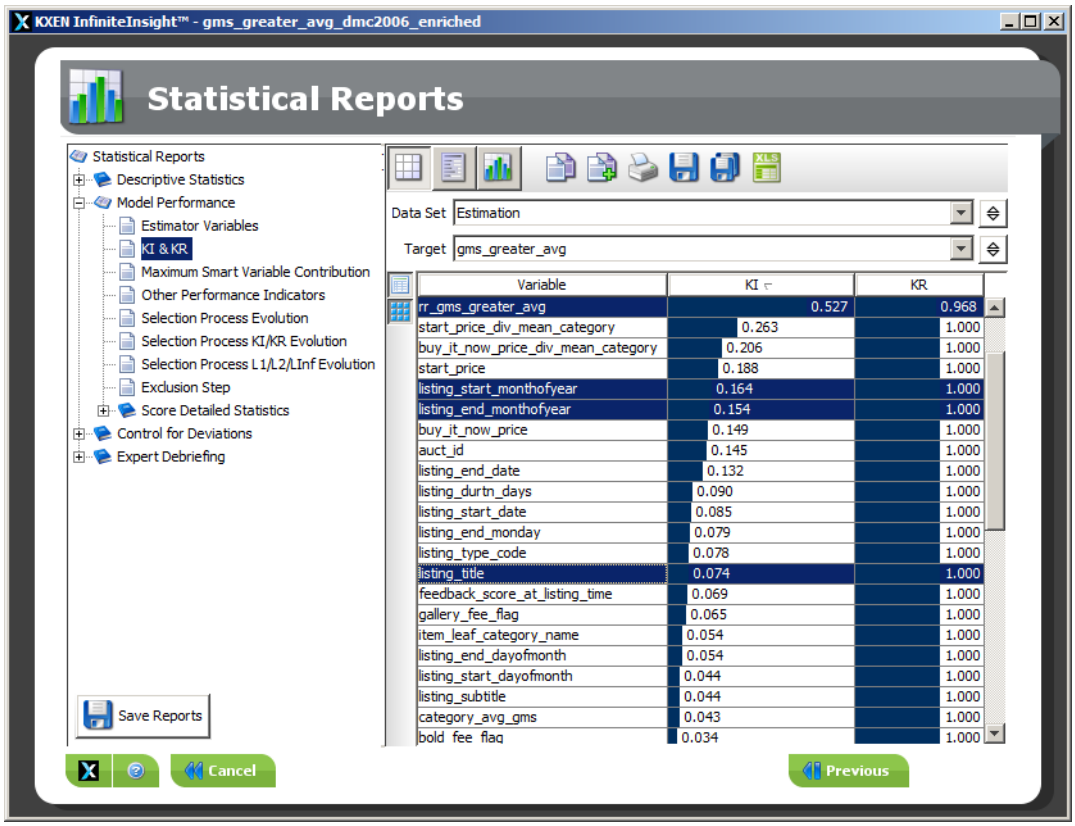
	KI	KR
Simple Method	0.468	0.970
Intermediate Method	**0.547**	**0.969**

The created variables give a better model. Indeed the KI has increased from 0.46 to 0.54 and the KR has also slightly increased.

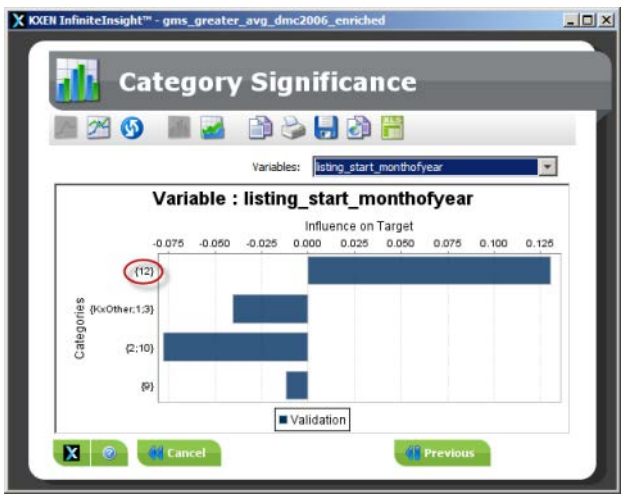
Adding data from already existing variables has led you to obtaining a model that has both a better quality and robustness.

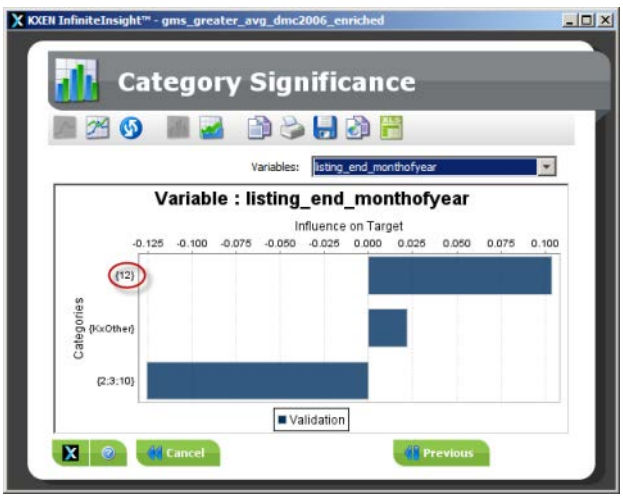
Taking a Closer Look at the Model

On the screen **Statistical Reports > Model Performance > KI & KR** (see below), you notice that the added variables have made a difference in the model since some appear among the variables with the higher individual KI. The individual KI represents the capacity of a variable to predict the target if only this variable was available.

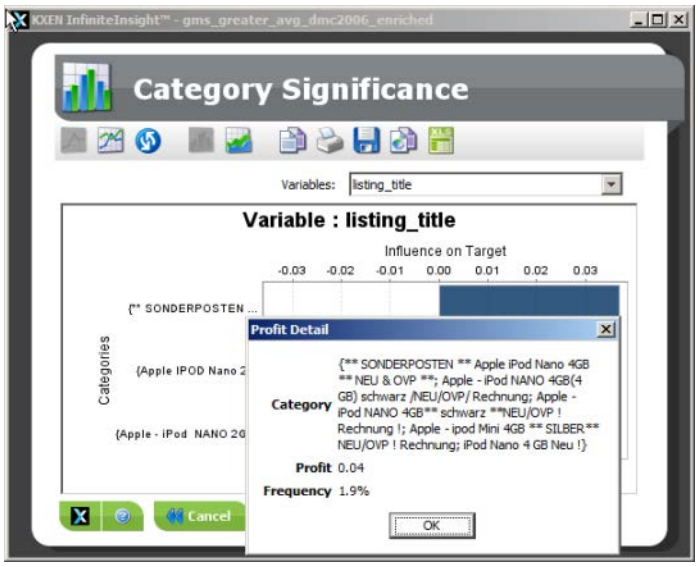


You can see that both variables **listing_start_monthofyear** and **listing_end_monthofyear** are appear in the top ten variables. When looking at their categories importance, you will notice that the auctions happening in December (indicated as **12** on the graph below) have a better chance to sale higher than the average. This can be explained by the fact that people buy more around Christmas than any other period of the year.





Another of the top variables that already appeared in the previous model is **listing_title**. When looking at the variable categories, you can see that each category contains many varied textual elements as shown in the screenshot below.



You can infer that this variable contains information that has yet to be exploited. Since this variable is a string, the best way to extract these hidden information is to use Text Coding, which leads you to the advanced method.

Advanced Method: Using Text Coding to Extract Information from the Textual Variables

Description

Although the intermediate method resulted in a model that was both accurate and robust, you still have textual data not yet exploited. Since *InfiniteInsight™ Modeler - Regression/Classification* is not designed to process such data, you will need to use a data encoding feature. That is where Text Coding comes into play! Text Coding is a data encoding feature that that allows building a representative vector of the textual entries; it splits texts in words unit and extracts roots from the dataset.

Text Coding is automatically included when 'textual' attributes are declared.

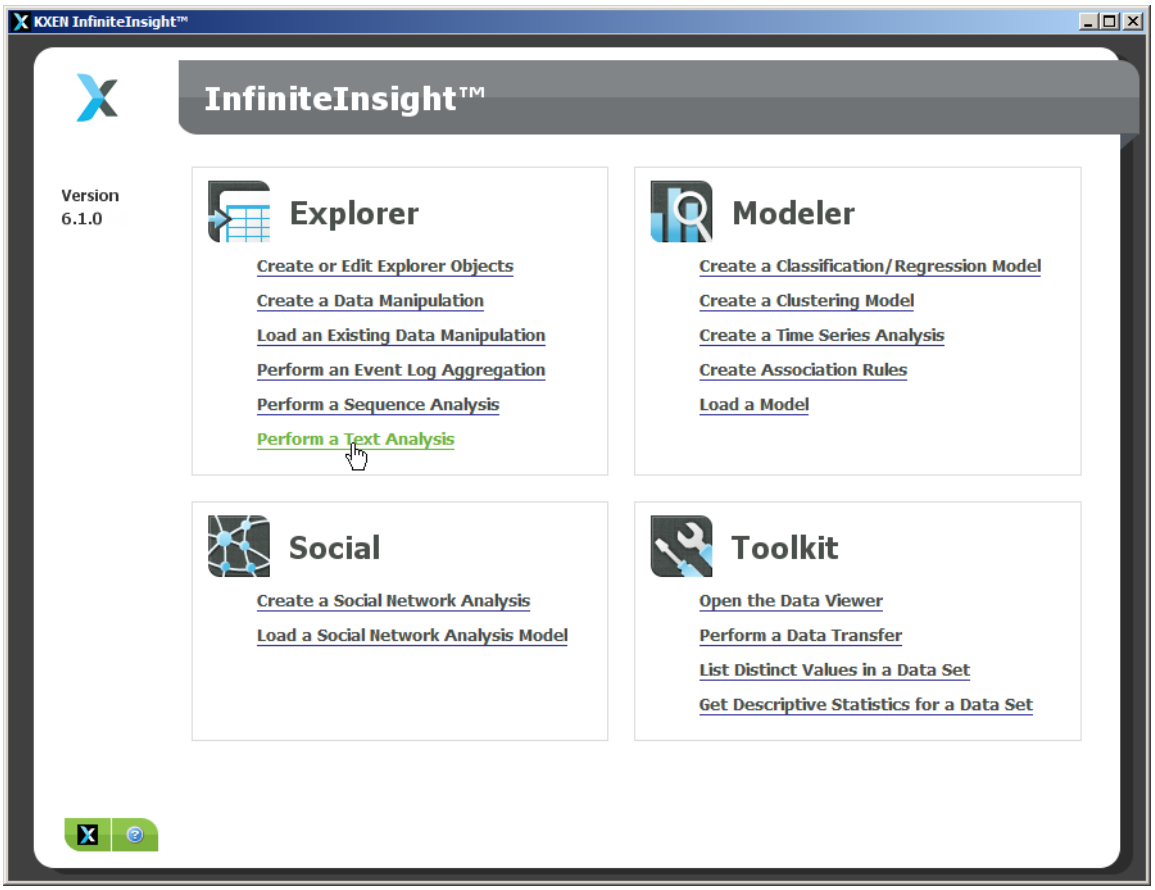
Modeling Process

Compared to using only *InfiniteInsight™ Modeler - Regression/Classification* as you did for the first two methods, using *InfiniteInsight™ Explorer - Text Coding* means performing the two additional steps below:

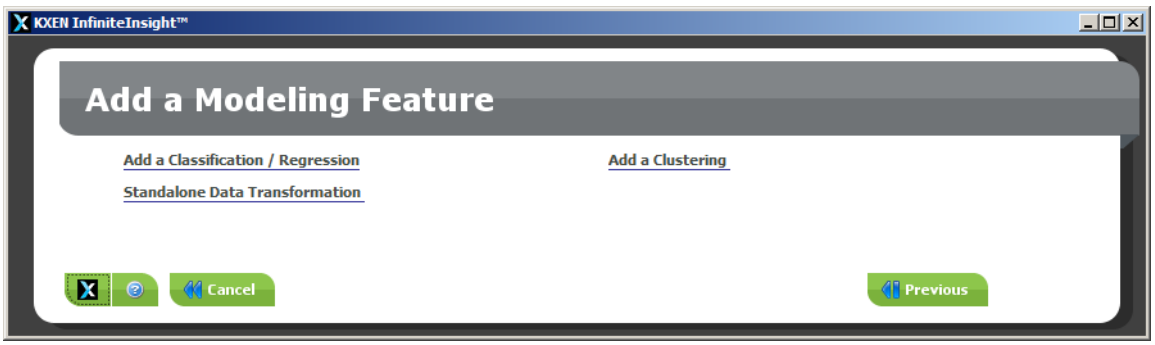
- Setting the language parameters
- Setting the dictionary and encoding parameters

Selecting the Type of Model to Create

1 In *InfiniteInsight™* main menu, select the option **Perform a Text Analysis** in the **Explorer** section.



The screen **Add a Modeling Feature** is displayed.



- **Add a Classification/Regression** analyzes the textual data, generates the corresponding variables and builds a Classification/Regression model on it
- **Add a Clustering** analyzes the textual data, generates the corresponding variables and builds a Clustering model on it
- **Standalobe Data Transformation** analyzes the textual data and generates the corresponding variables.

For this scenario

2 Click **Add a Classification/Regression**.

Summary of the Modeling Settings to Use

The table below summarizes the modeling settings you must use for the advanced method. Except for the Text Coding specific steps - which are grayed in the table below, the other settings are similar to the ones used for the intermediate method.

Text Coding steps are presented in details in the following sections.

For detailed procedures and more information, see the Modeling Process section of the Simple Method section.

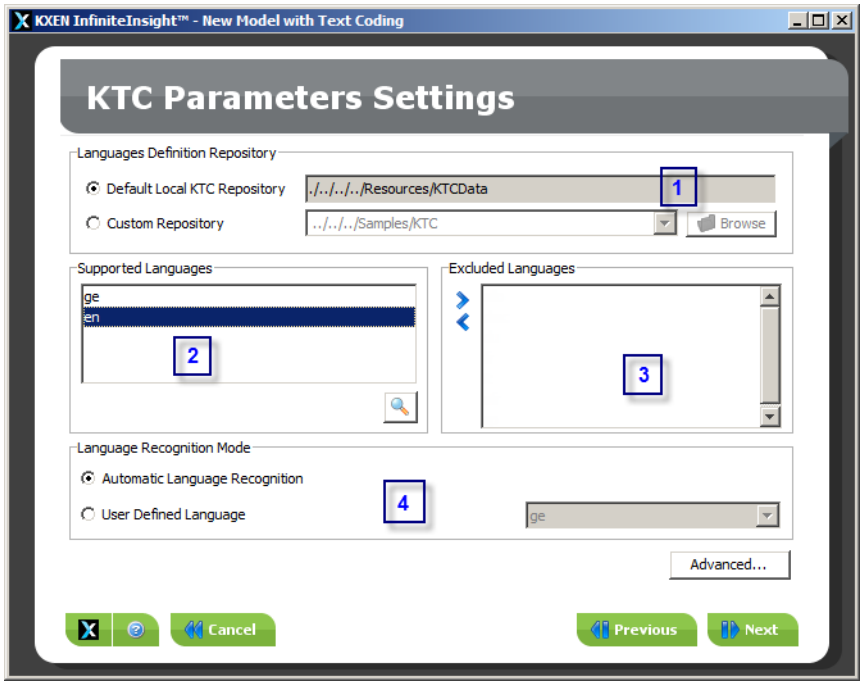
Task(s)	Screen	Settings
▪ Specifying the Data Source ▪ Selecting a Cutting Strategy	**Data to be Modeled**	▪ Select the option **Text Files** **in Data Type**. ▪ In the Folder field, select the folder **Samples/KTC/** ▪ In the Data Set field, select the file **dmc2006_enriched.txt.** ▪ Cutting strategy: **Random With No Test**
Describing the Data	**Data Description**	▪ Select **desc_dmc2006_enriched_textual.txt** as the description file. ▪ Check that **listing-title** is set as **textual**.
Text Coding - Setting the Language Definition	**KTC Parameters Settings**	▪ Keep **default** Language Definition Repository (blank) ▪ Select the **User Defined Language** option as the **Language Recognition Mode** ▪ Select **ge** (German) in the combo box as the User Defined Language
Text Coding - Setting the Dictionary and Encoding Parameters	**KTC Parameters Settings** (2)	▪ Keep the **default** settings
Selecting the Target Variable and a Weight Variable	**Selecting the Target Variable**	▪ Select **gms_greater_avg** as the target variable ▪ Do not select a weight variable
Selecting Explanatory Variables	**Selecting Variables**	▪ Exclude the variables **KxIndex** and **gms** from the list of variables to be used for modeling

Setting KTC Parameters

KTC Languages Parameters

The first panel **KTC Parameters Settings** allows you to chose the language settings:

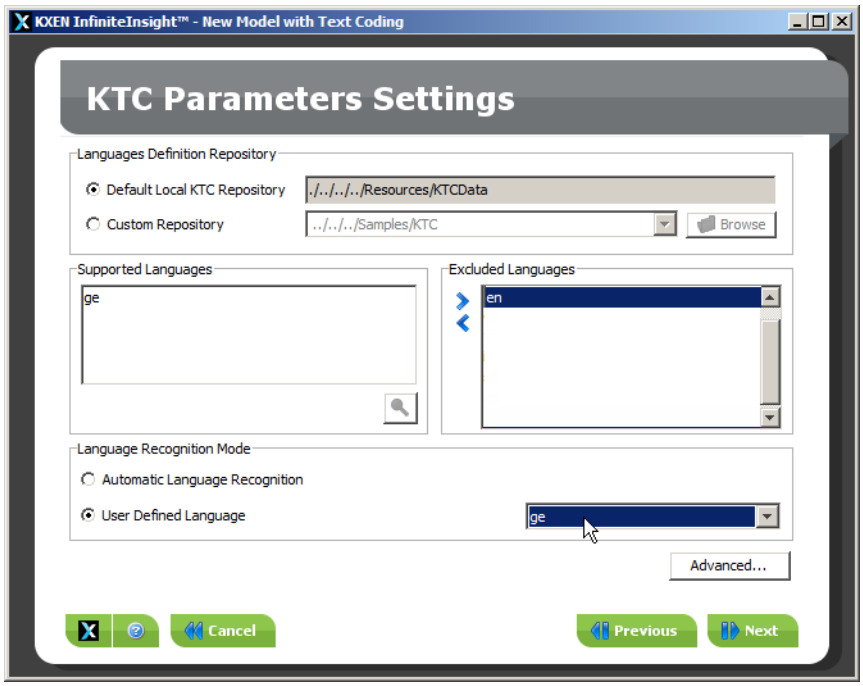
- Define the location of the **Language Definition Repository (1)**,
- Select the list of **Supported Languages (2)**,
- select the list of **Excluded Languages (3)**,
- Select the **Language Recognition Mode (4)**



For this Scenario

- Keep the default KTC Language Definition Repository (**./../../../Resources/KTCData**).
- You can exclude the language named **en** (for English).
- Select the **User Defined Language** option for the Language Recognition Mode
- If you did not exclude the English language, select **ge** (German) in the combo box as the User Defined Language.

The screen **should look like this**



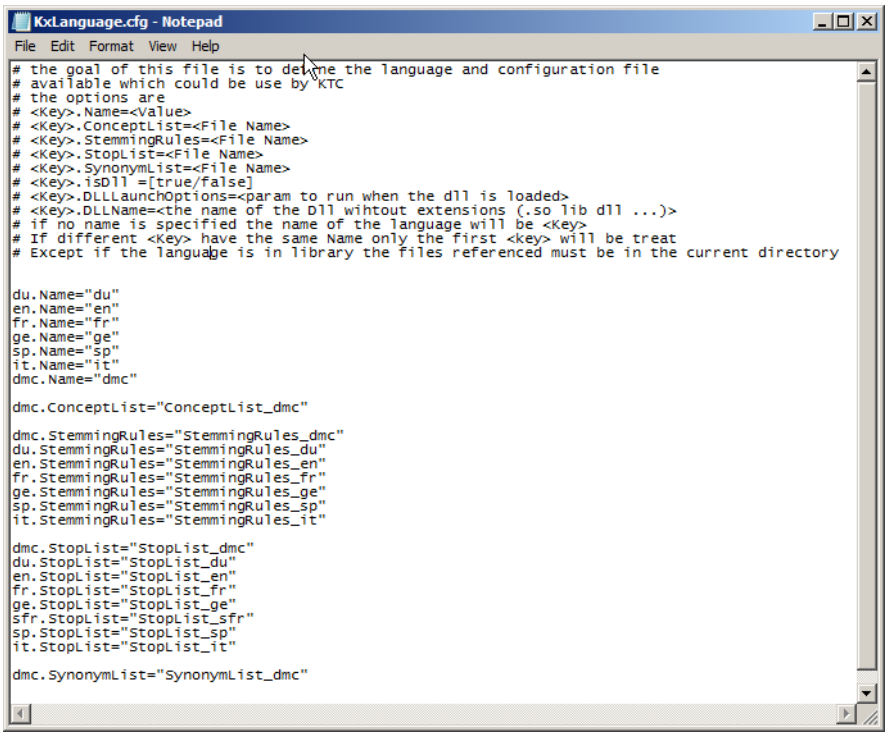
☑ For Advanced Users

You can create your own file to indicate the parameters on this panel. This file has to be named KxLanguage.cfg and needs to be structured as following:

```
<Key>.Name="<Value>"
<Key>.ConceptList="<File Name>"
<Key>.StemmingRules="<File Name>"
<Key>.StopList="<File Name>"
<Key>.SynonymList="<File Name>"
```

To add comments, begin these lines with `#.` `<Key>` refers to the defined language.

The configuration file KxLanguage.cfg should look like the following one:



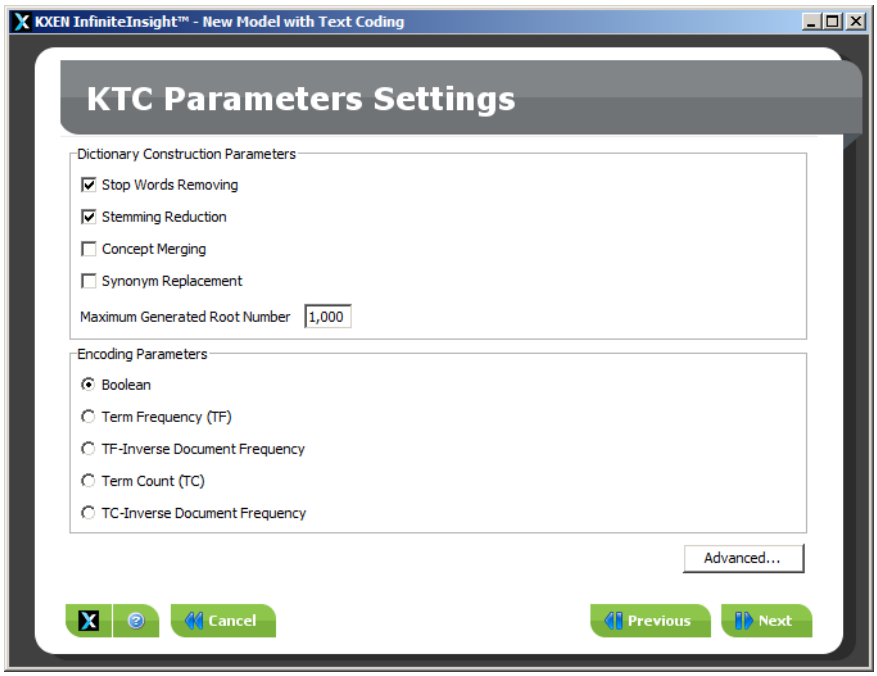
Notes

- the <Key> or language name has to be entered in the configuration file KxLanguage.cfg to be taken into account in the interface. If not set up there, the language will not appear in the interface.

- If no <Key> or language name is specified, the name of the language will be <Key>. If different <Key> have the same name, only the first <Key> will be treated. The referenced files have to be in the current directory.

KTC Dictionary and Encoding Parameters

The second panel **KTC Parameters Settings** allows you set the construction parameters for the dictionary and the encoding parameters.



Dictionary Construction Parameters

The dictionary is made of roots, that is meaningful words or terms. You can set the following parameters of the dictionary construction:

- **Stop Words Removing**: when this option is checked, the stop words are removed from the list of roots.
- **Stemming Reduction**: when this option is checked, the affixes are removed to limit the number of roots.
- **Concept Merging**: this option allows you to use an external file associating terms (that is groups of words designating a single concept, such as “the White House” or “credit card”) with concepts. Because it treats groups of words, this option is applied `before the removal of the` stop words and the stemming. You can create your own concepts dictionary by creating a text file named ConceptList_<LanguageCode> (without extension), which contains on each line a group of words associated with the corresponding concept. For example, you can create a concept list for an airline company:

```
word=concept
business-class=BusinessClass
first-class=FirstClass
flying-blue=FlyingBlue
```

Or you can apply the concept of “creditcard” to any credit card (such as “American Express”, “Visa Card”, …):

```
credit-card=creditcard
american-express=creditcard
visa-card=creditcard
mastercard=creditcard
```

Notes

- you have to put a **"=" sign of equality** between the words and the concepts, to replace the blanks (or every other separator) by **dashes** and to write the words in **lower-case letters** (since the concept merging is applied after the removal of all upper-case letters).
- you have to do the concept merging for the singular and plural forms of the words to cover all the occurrences.

The use of the concept list being language dependent, the appropriate list is automatically selected once the language has been either automatically identified, or selected by the user.

- **Synonyms Replacement**: this option allows you to use an external file defining synonymic roots. It will be used to replace some roots by a root selected by the user. This option is applied after the stop words have been removed and the stemming rules have been applied. You can create your own synonyms dictionary by creating a text file named `SynonymList_<LanguageCode>`, which contains on each line a root found by Text Coding associated with the synonym root as shown below:

```
<found_root>=<replacement_root>
```

The use of the synonyms list being language dependent, the appropriate list is automatically selected once the language has been either automatically identified, or selected by the user.

- **Maximum Generated Root Number**: this option allows you to select how many roots you want to keep in the dictionary. By default the roots with the highest frequencies are kept, but you can select a percentage of the most frequent roots to exclude by clicking the **Advanced** button.

Encoding Parameters

Each root is converted into a variable and when the root appears in a text, its presence can be encoded in three ways:

- **Boolean**: the presence of the word is encoded 1 and its absence is encoded 0.
- **Term Frequency**: the number of apparitions of the root in the current text.
- **TF-Inverse Document Frequency**: a measure of the general importance of a root in the current document relative to the whole set of documents based on Term Frequency.

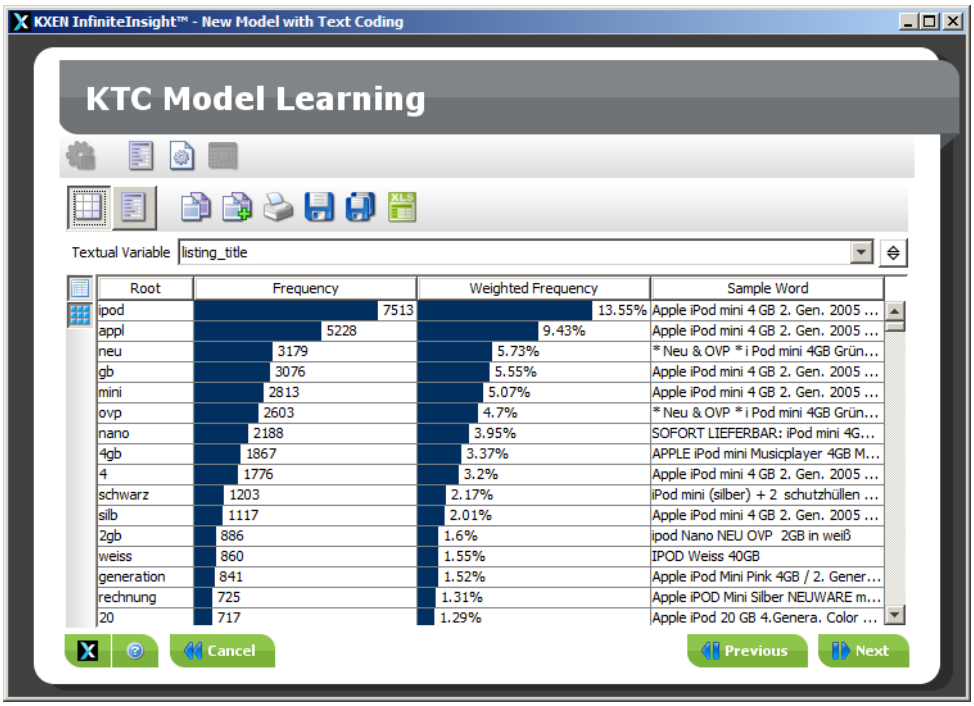
```
TF-IDF =
TF*log10(TotalNumberOfDocuments/NumberOfDocumentsContainingTheRoot)
```

- **Term Count**: the number of times the root appears in the current text.
- **TC-Inverse Document Frequency**: a measure of the general importance of a root in the current document relative to the whole set of documents based on Term Count.

```
TC-IDF =
TC*log10(TotalNumberOfDocuments/NumberOfDocumentsContainingTheRoot)
```

For this Scenario

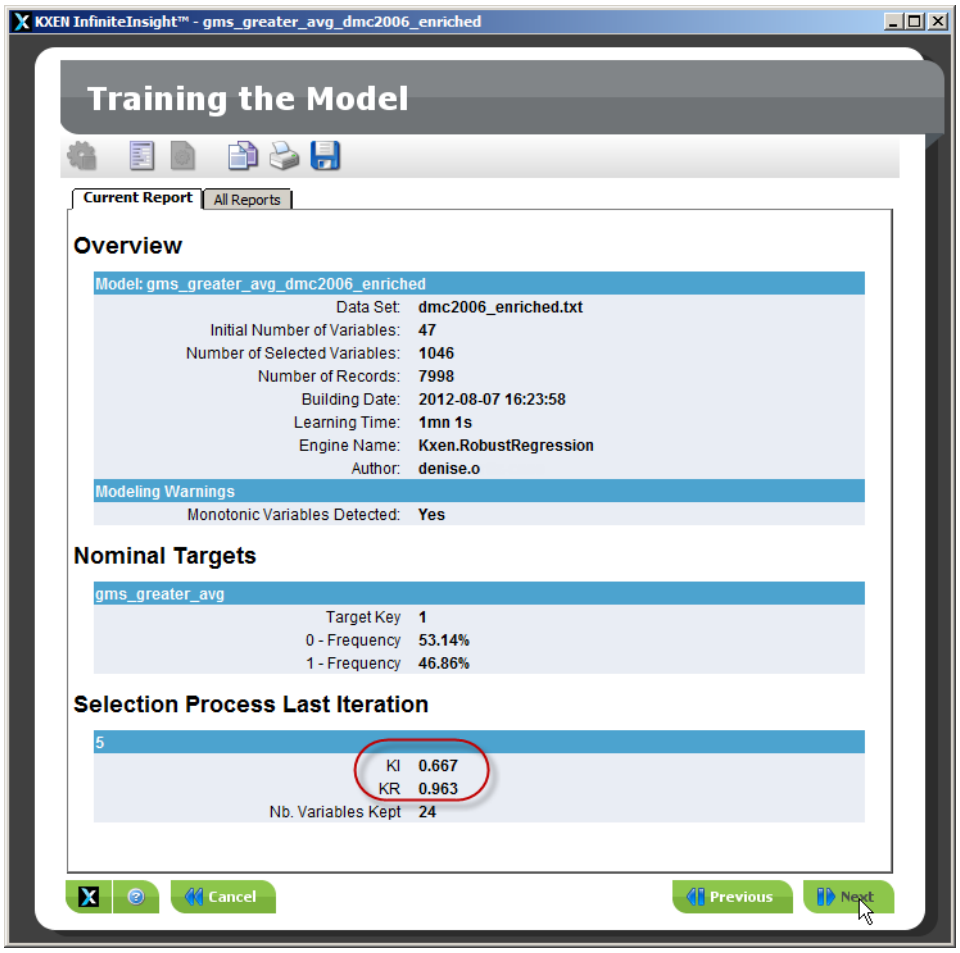
- Keep the default parameters
- Click the **Next** button, the panel **KTC Model Learning** is displayed.



This panel lists the roots identified by Text Coding in the analyzed textual variable, here **listing_title**, with their respective frequency of apparition in the data set. It allows you to identify the most frequent roots and to decide if these roots are really meaningful for your problem or not.

Results

The screen below shows the quality (KI) and robustness (KR) indicators obtained for the model generated with *InfiniteInsight™ Explorer - Text Coding*.



The table below compares these results with the ones obtained for the first two methods.

	KI	KR
Simple Method	0.468	0.970
Intermediate Method	0.547	0.969
Advanced Method	**0.667**	**0.963**

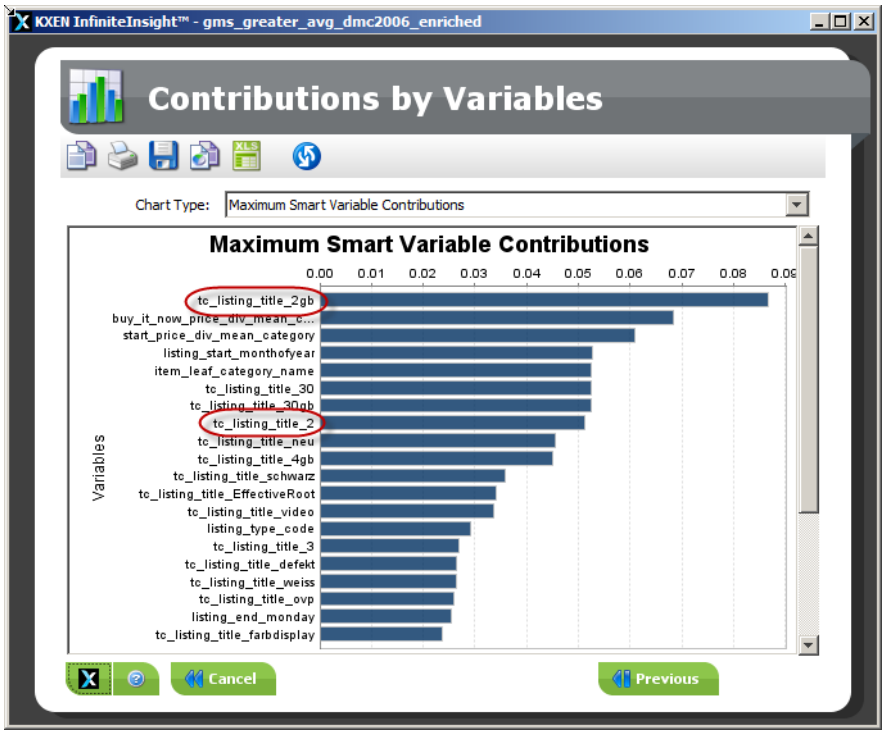
The analysis of textual variables gives a better model. Indeed the KI has increased from 0.55 to 0.66.

Using *InfiniteInsight™ Explorer - Text Coding* has led you to obtaining a model with a better quality and a high robustness.

Taking a Closer Look at the Model

On the screen **Contributions by Variables** (see below), you notice that the variables that have been created by Text Coding are important in the final model. For example **tc_listing_title_2gb** is the best maximum smart variable contribution. From this debriefing, you can see that 25 variables are displayed, 14 of which have been generated by Text Coding.

However after studying the roots listed in the panel **KTC Model Learning**, you can see that some of them are similar and should probably be merged. For example, both variables **tc_listing_title_2gb** and **tc_listing_title_2** exist and yet they contain the same information.



When building a model, Text Coding automatically generates two variables:

- **tc_<variable name>_EffectiveRoot**: this variable counts the final number of roots in the textual field csReferer
- **tc_<variable name>_CountInformation**: this variable counts the number of roots before filtering

Advanced Method without Stop Words and Stemming Rules

Description

In the results of the model using Text Coding, you can see that the variables created by Text Coding have brought information in the final model. For example **tc_listing_title_2gb** is the most contributive variable. You have seen that some of these variables contain the same information and should be grouped. However before grouping similar terms, you have to measure the impact of the German processing on the data set. To that effect, you will build a Text Coding model without specific German processing in order to see what its impact on the model quality is.

Modeling Process

The process of using Text Coding without German specific processing is approximately the same as the one you used for building the previous model. You will only need to change the dictionary and encoding parameters.

Summary of the Modeling Settings to Use

The table below summarizes the modeling settings you must use for the advanced method. The Text Coding specific steps are grayed in the table below and the step different from the previous model is indicated in blue. The other settings are similar to the ones used for the advanced method.

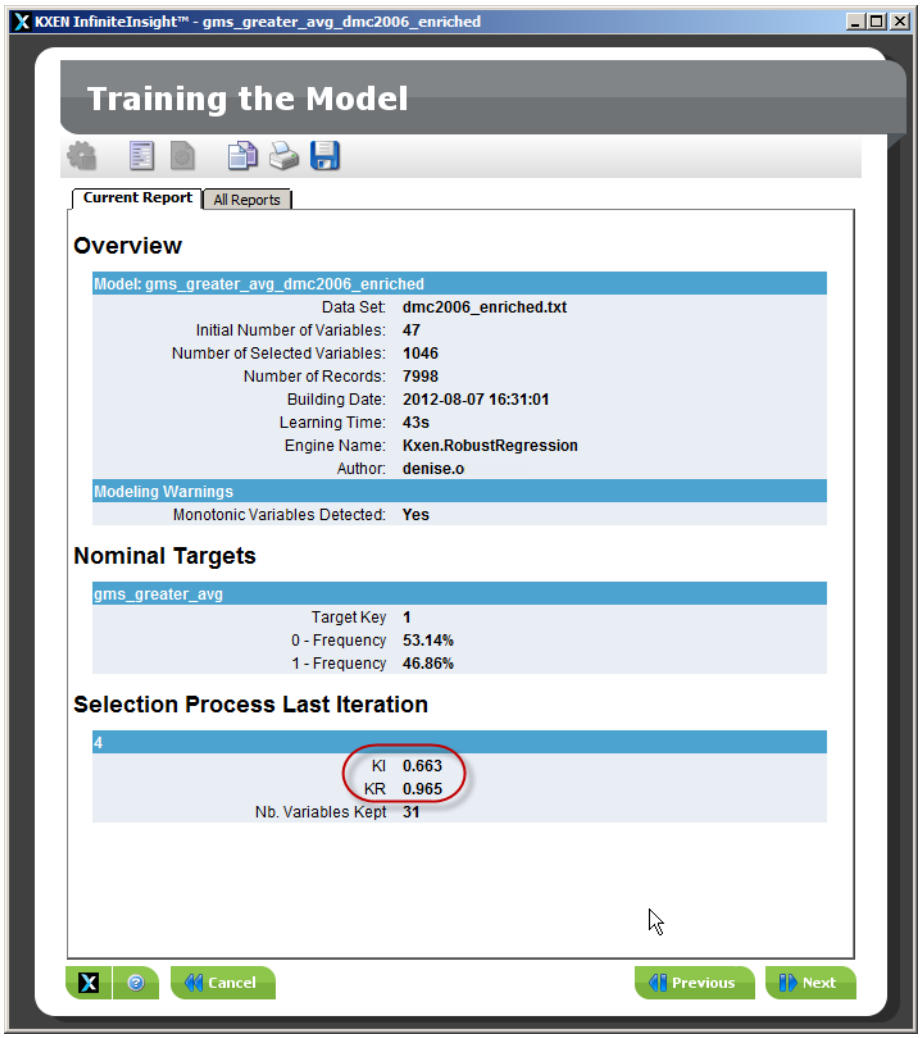
Text Coding steps are presented in details in the following sections.

For detailed procedures and more information, see the Modeling Process section of the Simple Method section.

Task(s)	Screen	Settings
▪ Specifying the Data Source ▪ Selecting a Cutting Strategy	**Data to be Modeled**	▪ Select the option **Text Files** in Data Type. ▪ In the Folder field, select the folder **Samples/KTC/** ▪ In the Data Set field, select the file **dmc2006**_enriched**.txt**. ▪ Cutting strategy: Random With No Test ▪
Describing the Data	**Data Description**	▪ Select desc_dmc2006_enriched_textual.txt as the description file. ▪ Check that listing-title is set as **textual**.
Text Coding - Setting the Language Definition	**KTC Parameters Settings**	▪ Keep **default** Language Definition Repository (./../../../Resources/KTCData) ▪ Select the **User Defined Language** option as the Language Recognition Mode ▪ Select **ge** (German) in the combo box as the User Defined Language
Text Coding - Setting the Dictionary and Encoding Parameters	**KTC Parameters Settings** (2)	▪ **Uncheck** Stop Word Removing ▪ **Uncheck** Stemming Reduction
Selecting the Target Variable and a Weight Variable	**Selecting the Target Variable**	▪ Select **gms_greater_avg** as the target variable ▪ Do not select a weight variable
Selecting Explanatory Variables	**Selecting Variables**	▪ Exclude the variables **KxIndex** and **gms** from the list of variables to be used for modeling

Results

The screen below shows the quality (KI) and robustness (KR) indicators obtained for the model generated with Text Coding.



The table below compares these results with the ones obtained for the first two methods.

	KI	KR
Simple Method	0.468	0.970
Intermediate Method	0.547	0.969
Advanced Method	0.667	0.963
Advanced Method without Stop Words and Stemming	**0.663**	**0.965**

There are not significant evolution of the KI and the KR. So you can conclude that using German stop words and stemming rules does not really add anything to the model.

Adapted Method: Defining a Specific Language for the Domain

Description

The results of disabling the German stop words and stemming rules show that they have no real impact on the model quality. Actually, after viewing the data, that makes sense. Indeed, the content of the **listing_title** variable can not be considered exactly as natural language but more as a language specific to a smaller domain.

So in this last method, you will define the stop words and stemming rules based on German but relevant to this domain only. This comes down to creating a specific language, which you will name **dmc**.

Modeling Process

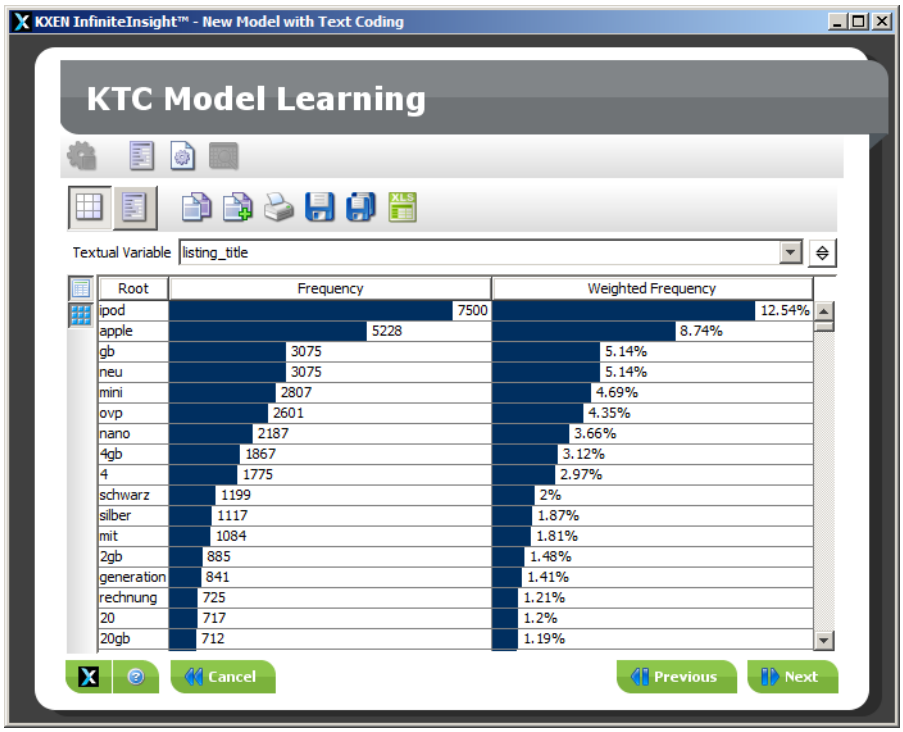
For this method you will have to create a list of stop words specific to the current domain and the stemming rules also adapted to this domain. The process is the same as the one you used for the advanced method, you will only need to set the language to the one you will create in the following steps. The two sections below describe how to create a stop words list and stemming rules. However since the process of creating the stop words list and the stemming rules can be lengthy, both are provided as an example in *InfiniteInsight™*. Thus the new language dmc will appear in the list of languages.

How to Detect Stop Words

Stop words are words that bring no information because they are too frequent or on the contrary that are less frequent.

Typically stop words are link words such as **aber**, **ob**, **ich**, **so**, **am**, **auf**… in German.

However other words can also be defined as stop words. The panel **KTC Model Learning** obtained by the advanced method without stop words and stemming rules can give you insight on which words can be considered as stop words.

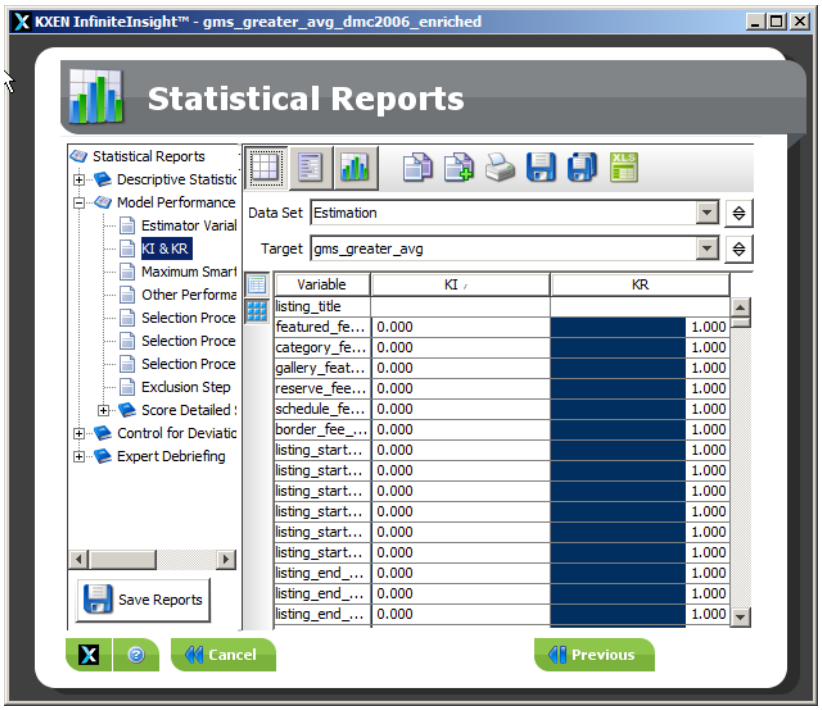


This panel lists for each textual variable:

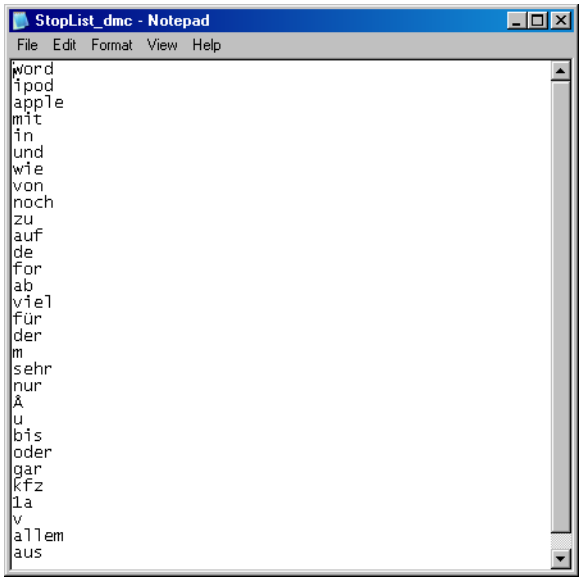
- the identified roots in the **Root** column
- the number of occurrences of each root in the **Frequency** column

Look at the roots **ipod** and **apple** for example. When you compare their number of occurrences with the total number of records in the data set, it appears that **ipod** is present 7500 times and **apple** 5228 times in a data set that counts 8000 lines. It is evident that they are much too frequent to contain information.

Another way to detect stop words is to use individual variable contribution after the Classification/Regression process in order to see the variables that have no KI.



The stop words list is stored in a text file named StopList_<language_code>. For example, the stop words list created for the specific language you are working on will be named **StopList_dmc**. The stop words list for the dmc language should look like the following file:



How to Build Simple Stemming Rules

According to the words displayed in the panel **KTC Model Learning**, you can build some simple stemming rules. Indeed the first thing that appears is that some words such as **20gb**, **20g** and **20** can be merged into a single words **20-GB**.

It can be defined by these stemming rules:

```
7       3       ^20gb$  nocond  nocond  ^20gb$  20-GB   4
8       3       ^20$    nocond  nocond  ^20$    20-GB   4
```

Note - See **Regular Expression Reminder** (on page 49) in the Annex.

The syntax of a stemming rule is the following:

```
Rule    Step    CondWord        CondR1  CondR2  Match   Replace StepAfter
```

The columns represent:

- `Rule`: the number of the rule
- `Step`: the step the rule belongs to
- `CondWord`: the condition applied to the word
- `CondR1`: the condition applied to the first region
- `CondR2`: the condition applied to the second region
- `Match`: the parts of the word to select for replacement
- `Replace`: the string to replace the matched part
- `StepAfter`: the step to go if the rule has been applied

So the stemming rule `7 3 ^20gb$ nocond nocond ^20gb$ 20-GB 4` says: "*if the word is 20gb then replace 20gb by 20-GB and go to stemming rules of step 4 if they exist*"

An other way to create stemming rules is to use the copy button in the panel **KTC Model Learning**, then paste the information on excel and sort the data by alphabetical order. Then you can identify different forms of the same words. For example, three different occurrences of a simple word can be identified:

- **eingeschweisst**
- **eingeschweist**
- **eingeschweißt**

So you can create two stemming rules to manage this word:

```
85 3 ^eingeschweist$ nocond nocond ^eingeschweist$ eingeschweisst 4
86 3 ^ eingeschweißt$ nocond nocond ^eingeschweißt$ eingeschweisst 4
```

These rules will replace two of the identified forms by the third one, so that only one form remains.

Moreover in the file you can find color names in different languages, for example **blau** in German and **blue** in English. So you can create the associated stemming rules such as:

```
65 3 ^blue$ nocond nocond ^blue$ blau 4
```

You can also create stemming rules that merge words that often appear together in the data set such as "original and packaging" which translate in German to:

- **original**
- **verpackt**

This can be managed by the following the stemming rules:

```
41      3       ^original$      nocond  nocond  ^original$      original_verpackt       4
42      3       ^verpackt$      nocond  nocond  ^verpackt$      original_verpackt       4
```

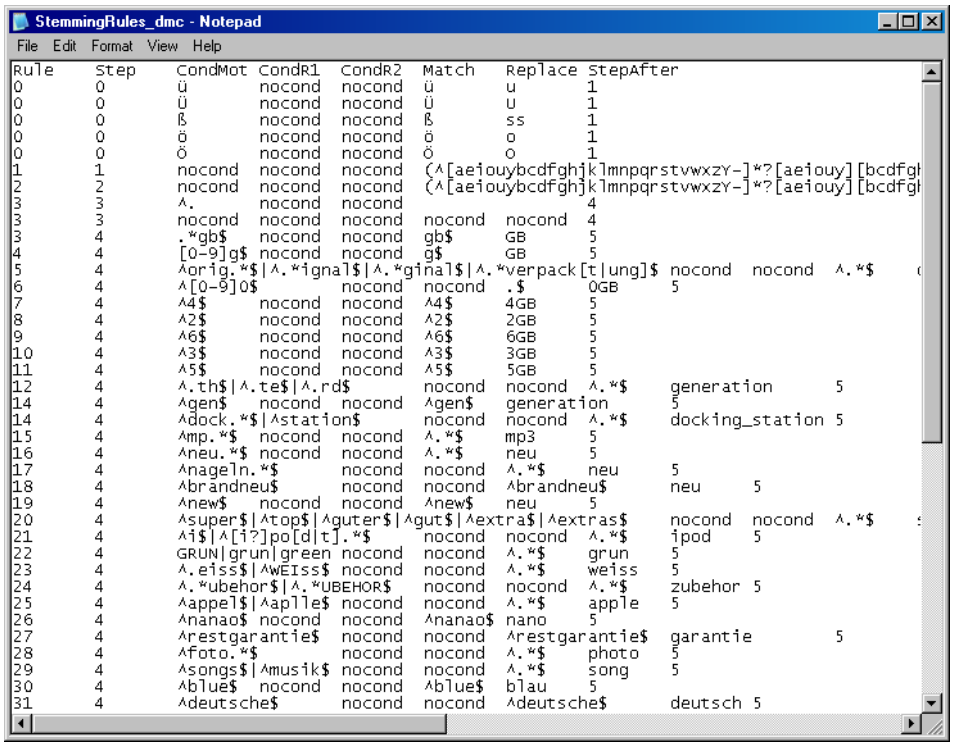
Lastly, you can merge correlated roots. In the previous model, look the **Variables Correlations** in **Statistic Report>Descriptive Statistics**.

Index	First Variable	Second Variable	Coefficient
11	listing_end_date	listing_start_date	0.995
3	buy_it_now_price	buy_it_now_price_div_mean_c...	0.986
18	start_price	start_price_div_mean_category	0.986
14	listing_end_monthofyear	listing_start_monthofyear	0.868
12	listing_end_date	listing_start_monthofyear	0.863
15	listing_start_date	listing_start_monthofyear	0.852
20	tc_listing_title_mp3	tc_listing_title_player	0.843
13	listing_end_monthofyear	listing_start_date	0.808
10	listing_end_date	listing_end_monthofyear	0.805
17	listing_type_code	start_price_div_mean_category	0.728
16	listing_type_code	start_price	0.721
4	category_avg_gms	item_leaf_category_name	0.621
7	item_leaf_category_name	tc_listing_title_30	0.59
5	category_avg_gms	tc_listing_title_30	0.57
9	item_leaf_category_name	tc_listing_title_video	0.55
6	category_avg_gms	tc_listing_title_video	0.541
19	tc_listing_title_guter	tc_listing_title_zustand	0.52
8	item_leaf_category_name	tc_listing_title_30gb	0.51
1	buy_it_now_listed_flag	buy_it_now_price	-0.962
2	buy_it_now_listed_flag	buy_it_now_price_div_mean_c...	-0.98

You can see that the roots **mp3** and **player** are highly correlated so you can create a stemming rule that will merge those roots into a single one.

```
43      3       ^mp3$   nocond  nocond  ^mp3$   mp3_player      4
44      3       ^player$        nocond  nocond  ^player$        mp3_player      4
```

The stemming rules are listed in a text file named StemmingRules_<language_code>. For example, the stemming rules created for the specific language you are working on will be sotred in the file **StemmingRules_dmc**. The Stemming Rules list for the dmc language should look like the following file:



Rule	Step	CondMot	CondR1	CondR2	Match	Replace	StepAfter
0	0	ü	nocond	nocond	ü	u	1
0	0	Ü	nocond	nocond	Ü	U	1
0	0	ß	nocond	nocond	ß	ss	1
0	0	ö	nocond	nocond	ö	o	1
0	0	Ö	nocond	nocond	Ö	O	1
1	1	nocond	nocond	nocond	(^[aeiouybcdfghjklmnpqrstvwxzY-]*?[aeiouy][bcdfgh		
2	2	nocond	nocond	nocond	(^[aeiouybcdfghjklmnpqrstvwxzY-]*?[aeiouy][bcdfgh		
3	3	^.	nocond	nocond		4	
3	3	nocond	nocond	nocond	nocond	nocond	4
3	4	.*gb$	nocond	nocond	gb$	GB	5
4	4	[0-9]g$	nocond	nocond	g$	GB	5
5	4	^orig.*$\|^.*ignal$|^.*ginal$\|^.*verpack[t\|ung]$	nocond	nocond	^.*$		
6	4	^[0-9]0$	nocond	nocond	.$	0GB	5
7	4	^4$	nocond	nocond	^4$	4GB	5
8	4	^2$	nocond	nocond	^2$	2GB	5
9	4	^6$	nocond	nocond	^6$	6GB	5
10	4	^3$	nocond	nocond	^3$	3GB	5
11	4	^5$	nocond	nocond	^5$	5GB	5
12	4	^.th$\|^.te$|^.rd$	nocond	nocond	^.*$	generation	5
14	4	^gen$	nocond	nocond	^gen$	generation	5
14	4	^dock.*$\|^station$	nocond	nocond	^.*$	docking_station	5
15	4	^mp.*$	nocond	nocond	^.*$	mp3	5
16	4	^neu.*$	nocond	nocond	^.*$	neu	5
17	4	^nageln.*$	nocond	nocond	^.*$	neu	5
18	4	^brandneu$	nocond	nocond	^brandneu$	neu	5
19	4	^new$	nocond	nocond	^new$	neu	5
20	4	^super$\|^top$|^guter$\|^gut$|^extra$\|^extras$	nocond	nocond	^.*$		
21	4	^i$\|^[i?]po[d\|t].*$	nocond	nocond	^.*$	ipod	5
22	4	GRUN|grun|green	nocond	nocond	^.*$	grun	5
23	4	^.eiss$\|^WEIss$	nocond	nocond	^.*$	weiss	5
24	4	^.*ubehor$\|^.*UBEHOR$	nocond	nocond	^.*$	zubehor	5
25	4	^appel$\|^apfle$	nocond	nocond	^.*$	apple	5
26	4	^nanao$	nocond	nocond	^nanao$	nano	5
27	4	^restgarantie$	nocond	nocond	^restgarantie$	garantie	5
28	4	^foto.*$	nocond	nocond	^.*$	photo	5
29	4	^songs$\|^musik$	nocond	nocond	^.*$	song	5
30	4	^blue$	nocond	nocond	^blue$	blau	5
31	4	^deutsche$	nocond	nocond	^deutsche$	deutsch	5

Summary of the Modeling Settings to Use

The table below summarizes the modeling settings you must use for the final method. The Text Coding specific steps are grayed in the table below and the steps different from the previous model are indicated in green. The other settings are similar to the ones used for the advanced method.

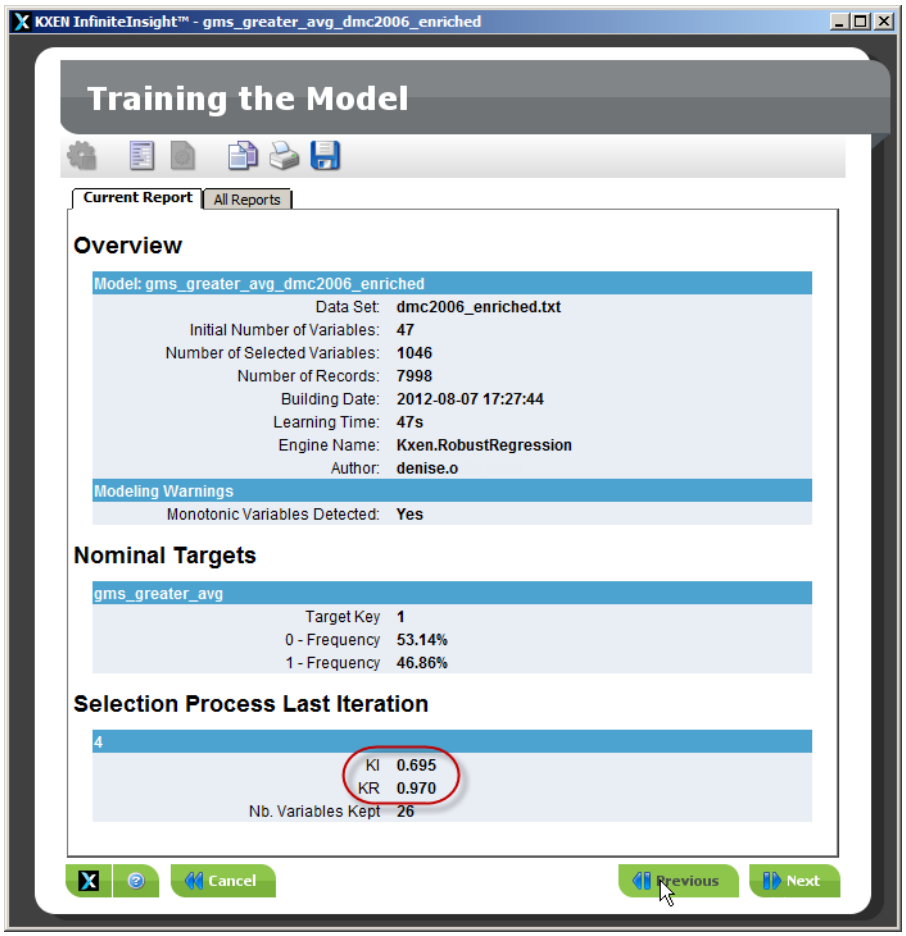
For detailed procedures and more information, see the Modeling Process section of the Simple Method section.

Task(s)	Screen	Settings
▪ Specifying the Data Source ▪ Selecting a Cutting Strategy	**Data to be Modeled**	▪ Select the option **Text Files** in Data Type. ▪ In the Folder field, select the folder **Samples/KTC/** ▪ In the Data Set field, select the file **dmc2006_enriched.txt**. ▪ Cutting strategy: **Random With No Test**
Describing the Data	**Data Description**	▪ Select **desc_dmc2006_enriched_textual.txt** as the description file. ▪ Check that listing-title is set as **textual**.
Text Coding - Setting the Language Definition	**KTC Parameters Settings**	▪ Keep **default** Language Definition Repository (./../../../Resources/KTCData) ▪ Select the **User Defined Language** option as the Language Recognition Mode ▪ Select **dmc** in the combo box as the User Defined Language
Text Coding - Setting the Dictionary and Encoding Parameters	**KTC Parameters Settings** (2)	▪ Check **Stop Word Removing** ▪ Check **Stemming Reduction** ▪ Check **Concept Merging** ▪ Check **Synonym Replacement**
Selecting the Target	**Selecting the Target**	▪ Select **gms_greater_avg** as the target variable

Task(s)	Screen	Settings
Variable and a Weight Variable	**Variable**	▪ Do not select a weight variable
Selecting Explanatory Variables	**Selecting Variables**	▪ Exclude the variables **KxIndex** and **gms** from the list of variables to be used for modeling

Results

The screen below shows the quality (KI) and robustness (KR) indicators obtained for the model generated with Text Coding adapted to the specific domain of application.



The table below compares these results with the ones obtained for the first two methods.

	KI	KR
Simple Method	0.468	0.970
Intermediate Method	0.547	0.969
Advanced Method	0.667	0.963

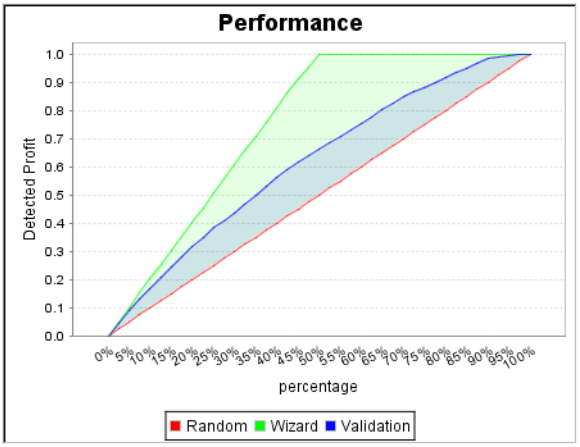
Advanced Method without Stop Words and Stemming	0.663	0.965
Adapted Method	**0.695**	**0.970**

The KI has largely improved and the KR stays very confident.

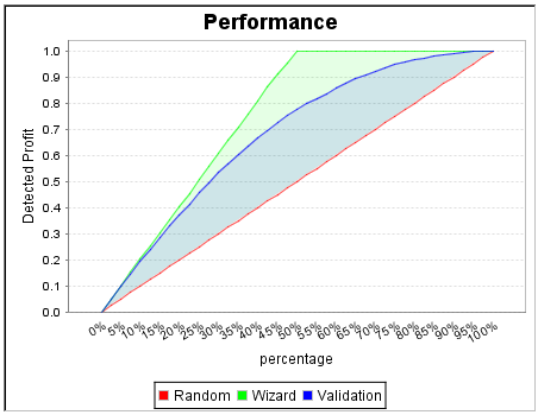
So you can see that from a simple Classification/Regression model performed on the original data set to a Text Coding + Classification/Regression model on improved data with a specialty language defined you have **gained a lot in model quality** (**+25%** in KI) without losing model robustness.

The increased quality of the model is clearly apparent on the model graphs below.

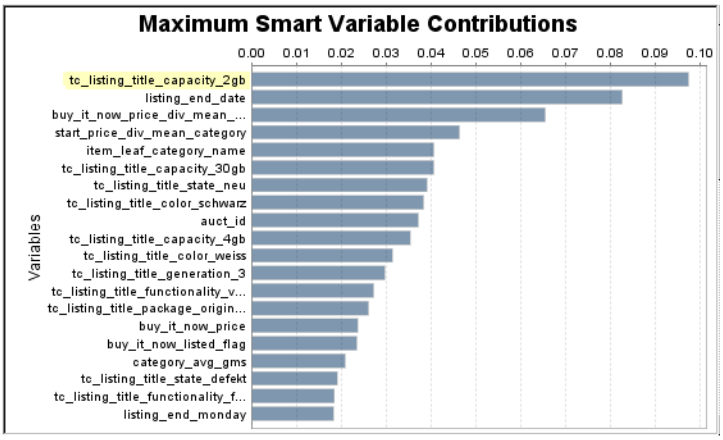
Model graph for the simple method



Model graph for the adapted method



With each method you have been able to uncover more and more information from your data. When looking at the Maximum Smart Variable Contributions below, you can see that the majority of the most contributive variables come from the textual analysis of the data. The variable that contributes the most to the target is **tc_listing_title_capacity_2gb**.

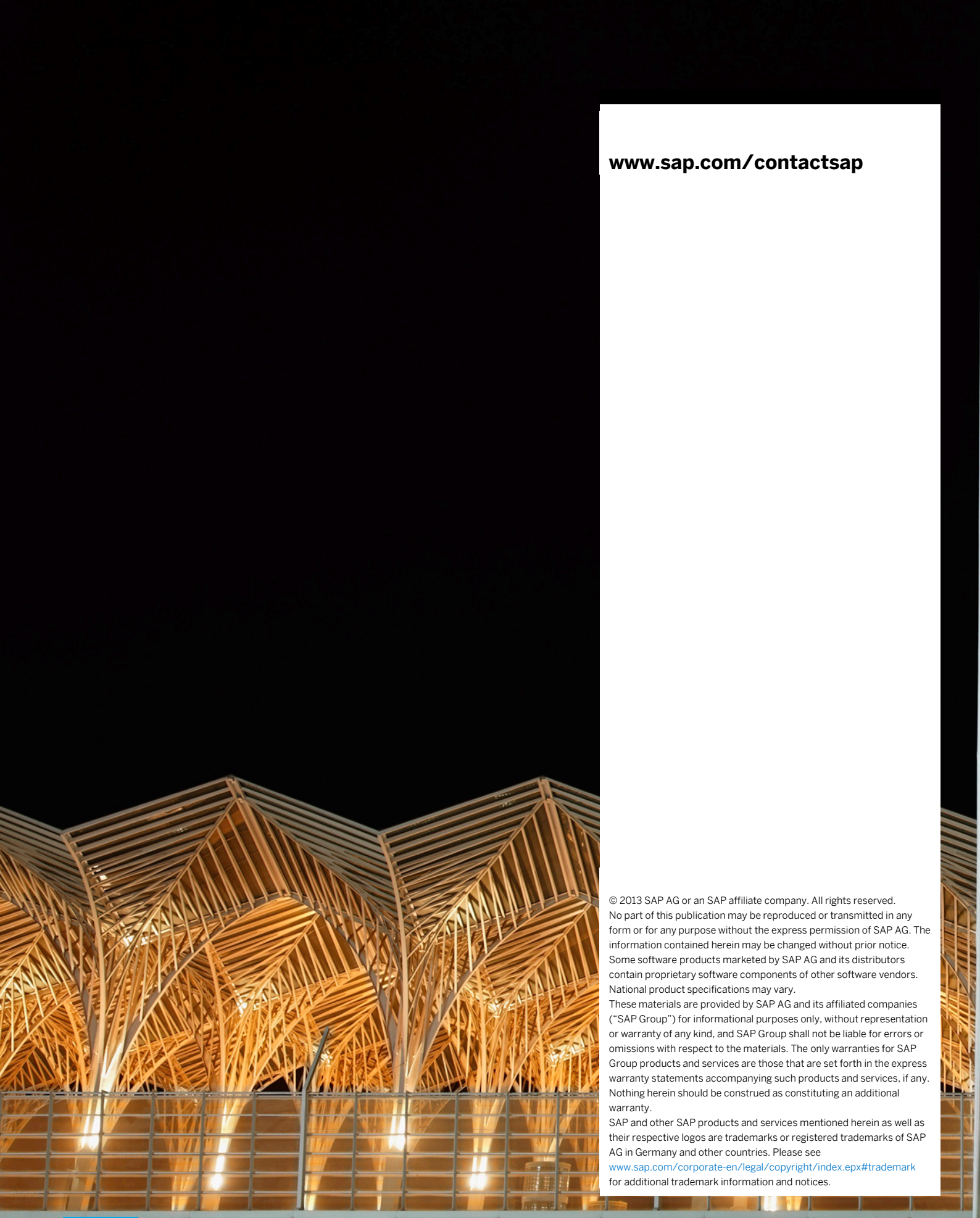


Annex

Regular Expression Reminder

The regular expressions engine used for the stemming rules is a PCRE engine (*Pearl Compatible Regular Expression*). The following table summarizes the main elements that can be used in the regular expressions:

\	general escape character with several uses
^	assert start of subject (or line, in multiline mode)
$	assert end of subject (or line, in multiline mode)
.	match any character except newline (by default)
[	start character class definition
]	end character class definition
|	start of alternative branch
(	start subpattern
)	end subpattern
?	extends the meaning of (, also 0 or 1 quantifier, also quantifier minimizer
*	0 or more quantifier
+	1 or more quantifier
{	start min/max quantifier
}	end min/max quantifier



www.sap.com/contactsap

© 2013 SAP AG or an SAP affiliate company. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.
Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.
National product specifications may vary.
These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.
SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries. Please see www.sap.com/corporate-en/legal/copyright/index.epx#trademark for additional trademark information and notices.